© 2026 Synopsys, Inc and/or ANSYS, Inc. All rights reserved.
Unauthorized use, distribution, or duplication is prohibited.

HFSS™ 3D Layout Scripting Guide

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2026 R1
January 2026

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001:2015 companies.

Copyright and Trademark Information

© 1986-2026 ANSYS, Inc. Unauthorized use, distribution or duplication is prohibited.

ANSYS, Ansys Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. All other brand, product, service and feature names or trademarks are the property of their respective owners. FLEXlm and FLEXnet are trademarks of Flexera Software LLC.

Disclaimer Notice

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. and ANSYS Europe, Ltd. are UL registered ISO 9001: 2015 companies.

U.S. Government Rights

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

Third-PartySoftware

See the legal information in the product help files for the complete Legal Notice for Ansys proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Table of Contents

Table of Contents **Contents-1**

1 - Introduction to Scripting **1-1**

Scripting Help Conventions 1-1

Variable Types 1-2

Introduction to IronPython 1-2

Scope 1-2

Python Compatibility 1-2

Advantages of IronPython 1-2

Scripting Using Iron Python 1-3

IronPython Script Execution Environment 1-4

Script Argument for IronPython 1-4

Scripting using Embedded VBScript or JavaScript 1-5

Scripting with Python 1-6

Standalone IronPython 1-7

Running Standalone IronPython 1-7

Using a Recorded Script 1-7

Creating an External Script 1-7

Example Script 1-8

IronPython Samples 1-9

Change property 1-10

Create a Cone using IronPython 1-11

Creating User Defined Primitives and User Defined Models in Python Scripts 1-14

Advantages Compared to C++ 1-15

Changes compared to C 1-15

Structures 1-15

Return Values for UDM and UDP Functions 1-15

Constants 1-15

Methods 1-15

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Output Parameters ... 1-16
Comparison with C function: ... 1-16
'List Size' Parameters ... 1-17
Comparison with C function: ... 1-17
Added Parameters ... 1-18
Developing a UDM/UDP ... 1-19
Creation ... 1-19
Location ... 1-19
Organize ... 1-20
Edit/Reload ... 1-20
UDPExtension ... 1-20
IUDPExtension Methods ... 1-20
Mandatory Methods ... 1-21
GetLengthParameterUnits() ... 1-21
Optional Methods ... 1-21
Example UDP ... 1-22
UDMExtension ... 1-23
Import ... 1-23
Main class: UDMExtension ... 1-23
IUDMExtension Methods ... 1-23
Mandatory Methods ... 1-23
Optional Methods ... 1-25
Example UDM ... 1-26
UDMFunctionLibrary ... 1-26
Functions list: ... 1-27
UDM/UDP Functions ... 1-28
Return Values for Each UDM and UDP Function ... 1-28
UDP/UDM Structures and Constants ... 1-30
UDP/UDM Structures ... 1-30
List of Structures ... 1-32

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UDP/UDM Constants ... 1-38
Enum constants: ... 1-38
UDP Python Example ... 1-40
Introduction to CPython ... 1-44
Creating an External Script ... 1-45
Start ansysedt as GRPC server ... 1-45
Connect Functions: ... 1-46
Example: ... 1-46
Launching Electronics Desktop ... 1-46
Connecting with a Running Instance of Electronics Desktop ... 1-47
Closing Electronics Desktop/Ending the Script ... 1-47
-grpcsrv Flag ... 1-47
Ansys Electronics Desktop Scripting ... 1-50
Overview of Electronics Desktop Scripting Objects ... 1-50
oAnsoftApp ... 1-51
oDesktop ... 1-51
oProject ... 1-51
oDesign ... 1-51
oEditor ... 1-51
oModule ... 1-52
Example Script Opening ... 1-52
GetActiveProject and GetActiveDesign for Wider Use ... 1-53
Running a Script ... 1-53
Within Electronics Desktop ... 1-53
From the Command Line ... 1-54
Recording a Script ... 1-54
Recording a Script to File ... 1-54
Recording a Script to a Project ... 1-55
Working with Project Scripts ... 1-56
Executing a Script from Within a Script ... 1-57

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Electronics Desktop Scripting Conventions ... 1-57
Named Arguments ... 1-57
IronPython Example ... 1-58
Setting Numerical Values ... 1-59
Layout Scripts and the Active Layer ... 1-59
Scripts and Locked Layers ... 1-60
PyAEDT (Beta) ... 1-60
2 - Object-Oriented Scripting ... 2-1
Object-Oriented Scripting Logic/Syntax ... 2-1
Basic Functions Overview ... 2-1
Body Properties and Modification ... 2-8
Retrieving Variables ... 2-8
Retrieving Datasets and Values ... 2-9
GetSolutionData API ... 2-10
Summary ... 2-11
Additional Details Specific to AEDT Solvers ... 2-14
Additional Details on Boundaries/Excitations ... 2-15
3D Component Encapsulation ... 2-16
Additional Details on Solve Setup ... 2-17
Object-Oriented Scripting for Editors ... 2-18
Schematic Editor ... 2-18
Object-Oriented Scripting for Materials ... 2-18
Example Change to Material Property Type: ... 2-22
Example Change to Vector Component Value: ... 2-23
Example Change to Choice Property Value: ... 2-24
Property Script Commands ... 2-25
Object Script Property Function Summary ... 2-27
Object Path ... 2-27
Property Object ... 2-27
Project Object ... 2-29

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Design Object 2-30
3D Modeler Object 2-30
Variable Object 2-31
Optimetrics Module Object: 2-32
Optimetrics Setup Object 2-33
ReportSetup(Results) Module Object: 2-33
ReportSetup(Results) Module Child Objects: 2-34
Radiation Module Object: 2-35
Radiation Module Child Objects: 2-35
Conventions Used in this Chapter 2-36
Callback Scripting Using PropHost Object 2-38
ChangeProperty (Schematic Editor and Layout Editor) 2-41
Property Array Structure 2-43
PropHost Functions 2-48
AddMenuProp 2-48
AddMenuProp2 2-49
AddProp 2-51
AddProp2 2-54
ExecuteScript 2-56
GetApplication 2-56
GetCallback 2-57
GetChangedProperty 2-58
GetDescription 2-58
GetDesign 2-59
GetEditor 2-59
GetEvaluatedText 2-60
GetFileName 2-60
GetHidden 2-61
GetProgress 2-62
GetPropHost 2-62

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropServers 2-62
GetPropTabType 2-63
GetReadOnly 2-63
GetRunStatus 2-64
GetTabTypeName 2-64
GetText 2-65
GetValue 2-66
IsValueConstant 2-66
PropertyExists 2-67
RemoveProp 2-67
SetCallback 2-68
SetDescription 2-69
SetHidden 2-69
SetReadOnly 2-70
SetText 2-70
SetValue 2-71
GetArrayVariables 2-77
GetProperties 2-78
GetPropertyValue 2-79
GetVariables 2-80
GetVariableValue 2-81
SetPropertyValue 2-81
SetVariableValue 2-83
3 - Application Object Script Commands 3-1
GetAppDesktop 3-2
4 - Desktop Object Script Commands 4-1
AddMessage 4-6
AreThereSimulationsRunning 4-7
ClearMessages 4-7
CloseAllWindows 4-11

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CloseProject ... 4-12
CloseProjectNoForce ... 4-12
Count ... 4-13
DeleteProject ... 4-14
DownloadJobResults ... 4-14
DeleteRegistryEntry ... 4-15
DoesRegistryValueExist ... 4-16
EnableAutoSave ... 4-17
ExportOptionsFiles ... 4-17
GetActiveProject ... 4-18
GetActiveScheduler ... 4-18
GetActiveSchedulerInfo ... 4-19
GetAutoSaveEnabled ... 4-20
GetBuildDateTimeString ... 4-21
GetCustomMenuSet ... 4-21
GetDefaultUnit ... 4-22
GetDesktopConfiguration ... 4-24
GetDistributedAnalysisMachines ... 4-24
GetDistributedAnalysisMachinesForDesignType ... 4-25
GetExeDir ... 4-26
GetGDIObjectCount ... 4-26
GetLibraryDirectory ... 4-27
GetLocalizationHelper ... 4-27
GetMessages ... 4-28
GetMonitorData ... 4-29
GetPersonalLibDirectory ... 4-30
GetProcessID ... 4-30
GetProjectDirectory ... 4-31
GetProjectList ... 4-31
GetProjects ... 4-32

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetRegistryInt 4-33
GetRegistryString 4-33
GetRunningInstancesMgr 4-34
GetSchematicEnvironment 4-35
GetScriptingToolsHelper 4-35
GetSysLibDirectory 4-36
GetTempDirectory 4-37
GetUserLibDirectory 4-37
GetVersion 4-38
IsFeatureEnabled 4-38
KeepDesktopResponsive 4-39
LaunchJobMonitor 4-40
NewProject 4-41
OpenAndConvertProject 4-41
OpenProject 4-42
OpenProjectWithConversion 4-43
PageSetup (Layout Editor) 4-43
PauseRecording 4-44
PauseScript 4-44
Print 4-45
QuitApplication 4-45
RefreshJobMonitor 4-46
ResetLogging 4-47
RestoreProjectArchive 4-47
RestoreWindow 4-48
ResumeRecording 4-49
RunACTWizardScript 4-49
RunProgram 4-50
RunScript 4-50
RunScriptWithArguments 4-52

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SelectScheduler 4-53
SetActiveProject 4-54
SetActiveProjectByPath 4-55
SetCustomMenuSet 4-55
SetDesktopConfiguration 4-56
SetLibraryDirectory 4-57
SetProjectDirectory 4-58
SetRegistryFromFile 4-58
SetRegistryInt 4-59
SetRegistryString 4-60
SetSchematicEnvironment 4-60
SetTempDirectory 4-61
ShowDockingWindow 4-62
Sleep 4-62
StopSimulations 4-63
SubmitJob 4-64
TileWindows 4-65
Desktop Commands For Registry Values 4-65
DeleteRegistryEntry 4-66
DoesRegistryValueExist 4-67
GetRegistryInt 4-68
GetRegistryString 4-68
SetRegistryFromFile 4-69
SetRegistryInt 4-70
SetRegistryString 4-70
ImportExport Tool Commands 4-71
ImportANF 4-72
ImportANFv2 4-72
ImportAutoCAD 4-73
ImportAWRMicrowaveOffice 4-74

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportEDB ... 4-75
ImportExtracta ... 4-75
ImportGDSII ... 4-76
ImportGerber ... 4-77
ImportIDF ... 4-78
ImportIDFandMerge ... 4-81
ImportIDX ... 4-82
ImportIPC ... 4-84
ImportODB ... 4-85
ImportXFL ... 4-85
5 - Running Instances Manager Script Commands ... 5-1
GetAllRunningInstances ... 5-1
GetRunningInstanceByProcessID ... 5-1
GetRunningInstancesMgr ... 5-2
6 - Project Object Script Commands ... 6-1
AddDataset ... 6-4
AddMaterial ... 6-6
AnalyzeAll [project] ... 6-9
ChangeProperty ... 6-10
ClearMessages ... 6-13
CloneMaterial ... 6-14
Close ... 6-15
CopyDesign ... 6-15
CutDesign ... 6-16
DeleteDataset ... 6-17
DeleteDesign ... 6-17
DeleteToolObject ... 6-18
EditDataset ... 6-18
EditMaterial ... 6-20
ExportDataset ... 6-27

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportMaterial 6-28
GetActiveDesign 6-28
GetArrayVariables 6-29
GetChildNames [Project] 6-30
GetChildObject [Project] 6-30
GetChildTypes [Project] 6-31
GetDefinitionManager 6-32
GetDependentFiles 6-32
GetDesign 6-33
GetDesigns 6-33
GetEDBHandle 6-34
GetLegacyName 6-35
GetName [Project] 6-35
GetObjPath [Project] 6-36
GetPath 6-36
GetPropEvaluatedValue 6-37
GetPropNames [Project] 6-38
GetPropSIValue 6-39
GetPropValue [Project] 6-40
GetProperties 6-40
GetPropertyValue 6-41
GetTopDesignList 6-43
GetVariableValue 6-43
GetVariables 6-44
ImportDataset 6-45
Note About File Types: 6-45
InsertDesign 6-46
InsertDesign (Layout Editor) 6-47
InsertDesignWithWorkflow 6-48
InsertToolObject 6-49

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Paste (Project Object) 6-49
Redo [Project Level] 6-50
RemoveAllUnusedDefinitions 6-50
RemoveMaterial 6-51
RemoveUnusedDefinitions 6-51
Rename 6-52
RunToolkit 6-53
Save 6-54
SaveAs 6-54
SaveAsStandAloneProject 6-56
SaveProjectArchive 6-57
SetActiveDefinitionEditor 6-58
SetActiveDesign 6-58
SetPropValue [Project] 6-59
SetPropertyValue 6-60
SetVariableValue 6-61
SimulateAll 6-62
Undo [Project] 6-62
UpdateDefinitions 6-63
7 - Design Object Script Commands 7-1
AddDataset 7-7
AddDynamicLink 7-8
AddModelingProperties 7-10
Analyze 7-10
AnalyzeAll [design] 7-11
AnalyzeAllNominal 7-11
ApplyMeshOps 7-12
ChangeProperty[ReportSetup] 7-12
CopyItemCommand 7-19
CreateReport 7-19

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteDataset 7-32
DeleteDesignInstance 7-33
DeleteFieldVariation 7-33
DeleteFullVariation 7-34
DeleteOutputVariable 7-35
DeleteSolutionVariation 7-35
EditCoSimulationOptions 7-36
EditDataset 7-37
EditImportData 7-39
EditInfiniteArray 7-39
Edit (Layout Editor) 7-40
EditLayoutForLayoutComponent 7-41
EditNotes 7-42
EditNotes (Layout Editor) 7-42
EditOptions (Layout Editor) 7-43
EditOutputVariable 7-43
EMDesignOptions 7-44
ExportCircuit (Layout Editor) 7-47
ExportConvergence 7-52
ExportDataset 7-52
ExportForHSpice 7-53
ExportForHSpice (Layout Editor) 7-54
ExportForSpice (Layout Editor) 7-56
ExportMeshStats 7-57
ExportNetworkData 7-58
ExportNetworkData 7-60
ExportNMFData 7-62
ExportProfile 7-63
ExportReport 7-64
ExportRLGCData 7-65

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Generate Mesh 7-67
GetChildNames [Design] 7-68
GetChildObject [Design] 7-68
GetChildTypes [Design] 7-69
GetDesignID 7-69
GetDesignName 7-70
GetDesignType 7-71
GetDesignValidationInfo 7-71
GetEditor 7-72
GetEditor (Layout Editor) 7-72
GetGeometryIdsForNetLayerCombinations 7-73
GetGeometryIdsForAllNetLayerCombinations 7-74
GetManagedFilesPath 7-75
GetModule 7-75
GetModule (Layout Editor) 7-76
GetModule (RadiationSetupMgr) 7-77
GetName 7-78
GetName (Layout Editor) 7-79
GetNoteText 7-79
GetObjPath [Design] 7-80
GetOutputVariableValue 7-81
GetOutputVariables 7-82
GetProject 7-82
GetPropHost 7-83
GetProperties 7-83
GetPropertyValue 7-84
GetPropNames [Design] 7-86
GetPropValue [Design] 7-86
GetVariables 7-87
GetVariableValue 7-88

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportDataset ... 7-88
Note About File Types: ... 7-89
InsertDesign ... 7-90
OverlayCurrents (Layout Editor) ... 7-91
OverlayFarField (Layout Editor) ... 7-92
OverlayMesh (Layout Editor) ... 7-92
OverlayNearField (Layout Editor) ... 7-92
PasteDesign (Design Object) ... 7-93
PasteItemCommand (Layout Editor) ... 7-94
PushExcitations ... 7-94
Redo [Design] ... 7-96
Redo (Layout Editor) ... 7-97
RemoveImportData (Layout Editor) ... 7-97
RemoveModelingProperties (Layout Editor) ... 7-97
RenameDesignInstance ... 7-98
RenameDesignInstance (Layout Editor) ... 7-98
RenameImportData (Layout Editor) ... 7-99
RenameSource [Interface Source] ... 7-99
ReportTemplates (Layout Editor) ... 7-100
RunToolkit ... 7-100
SetActiveEditor ... 7-101
SetActiveEditor (Layout Editor) ... 7-101
SetBendGraphicsMode ... 7-102
SetDesignMode ... 7-102
SetDoMeshAssembly ... 7-103
SetFastTransformationForLayoutComponent ... 7-109
SetObjectAttributesForLayoutComponent ... 7-110
SetPropValue [Design] ... 7-112
SetPropertyValue ... 7-113
SetShowLayoutForLayoutComponent ... 7-114

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetVariableValue 7-115
Undo [Design] 7-115
Undo (Layout Editor) 7-116
ValidateCircuit (Layout Editor) 7-116
8 - 3D Modeler Editor Script Commands 8-1
Conventions Used in this Chapter: 8-1
<AttributesArray> 8-1
<SelectionsArray> 8-2
Draw Menu Commands 8-4
Create3DComponent 8-6
CreateBondwire 8-11
CreateBox 8-14
CreateCircle 8-16
CreateCone 8-18
CreateCutplane 8-20
CreateCylinder 8-22
CreateEllipse 8-24
CreateEquationCurve 8-26
CreateEquationSurface 8-29
CreateHelix 8-31
CreatePoint 8-33
CreatePolyline 8-34
CreateRectangle 8-37
CreateRegion 8-40
CreateRegularPolygon 8-42
CreateRegularPolyhedron 8-44
CreateSphere 8-46
CreateSpiral 8-48
CreateTorus 8-49
CreateUserDefinedPart 8-51

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit3DComponent ... 8-56
Edit3DComponentDefinition ... 8-58
EditPolyline ... 8-60
Get3DComponentDefinitionNames ... 8-63
Get3DComponentInstanceNames ... 8-64
Get3DComponentMaterialNames ... 8-64
Get3DComponentMaterialProperties ... 8-65
Get3DComponentParameters ... 8-65
Get3DComponentPartNames ... 8-66
Insert3DComponent ... 8-66
InsertComponent ... 8-68
InsertPolylineSegment ... 8-69
SweepAlongPath ... 8-71
SweepAlongVector ... 8-72
SweepAroundAxis ... 8-74
SweepFacesAlongNormal ... 8-75
SweepFacesAlongNormalWithAttributes ... 8-77
UpdateComponentDefinition ... 8-78
Edit Menu Commands ... 8-79
Copy ... 8-80
DeletePolylinePoint ... 8-81
DuplicateAlongLine ... 8-81
DuplicateAroundAxis ... 8-83
DuplicateMirror ... 8-85
Mirror ... 8-87
Move ... 8-89
OffsetFaces ... 8-90
Paste (Model Editor) ... 8-91
Rotate ... 8-92
Scale ... 8-93

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Modeler Menu Commands 8-94
AlignFaces 8-96
AssignMaterial 8-97
Chamfer 8-98
CleanUpModel 8-100
Connect 8-100
CoverLines 8-101
CoverSurfaces 8-102
CreateEntityList 8-102
CreateFaceCS 8-104
CreateGroup 8-107
CreateNamedSelection 8-109
CreateObjectCS 8-110
CreateObjectFromEdges 8-114
CreateObjectFromFace 8-116
CreateObjectFromFaces 8-117
CreateRelativeCS 8-118
DeleteEmptyGroups 8-120
DeleteLastOperation 8-121
DeleteOperation 8-121
DetachEdges 8-123
DetachFaces 8-124
EditEntityList 8-125
EditFaceCS 8-126
EditNamedSelection 8-130
EditObjectCS 8-131
EditRelativeCS 8-136
Export 8-137
ExportModelImageToFile 8-139
ExportModelMeshToFile 8-142

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Fillet 8-143
FlattenGroup 8-144
GenerateHistory 8-145
HealObject 8-146
Import 8-148
ImportDXF [Modeler] 8-151
ImportFromClipboard 8-153
ImportGDSII [Modeler] 8-154
Imprint 8-156
ImprintProjection 8-157
Intersect 8-159
MoveCStoEnd 8-160
MoveEntityToGroup 8-160
MoveFaces 8-161
ProjectSheet 8-163
PurgeHistory 8-164
ReplaceWith3DComponent 8-164
Section 8-166
SeparateBody 8-168
SetModelUnits 8-168
SetWCS 8-169
ShowWindow 8-170
Simplify 8-171
Split 8-173
Stitch 8-175
Subtract 8-176
SweepFacesAlongNormal 8-177
ThickenSheet 8-178
UncoverFaces 8-179
Unite 8-180

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Ungroup ... 8-182
WrapSheet ... 8-182
Other oEditor Commands ... 8-183
AddDefinitionFromLibFile ... 8-186
AssignSurfaceMaterial ... 8-188
BreakUDMConnection ... 8-190
ChangeProperty ... 8-191
CloseAllWindows[Editor] ... 8-195
Delete ... 8-195
FitAll ... 8-196
GenerateAllUserDefinedModels ... 8-196
GenerateUserDefinedModel ... 8-197
GeometryCheckAndAutofix ... 8-197
GetBodyNamesByPosition ... 8-199
GetChildNames [Modeler] ... 8-200
GetChildObject [Modeler] ... 8-202
GetChildTypes [Modeler] ... 8-204
GetEdgeByPosition ... 8-205
GetEdgeIDsFromFace ... 8-207
GetEdgeIDsFromObject ... 8-207
GetEdgeLength ... 8-208
GetEntityIDsContainedByNamedSelection ... 8-208
GetEntityListIDByName ... 8-209
GetFaceArea ... 8-209
GetFaceCenter ... 8-210
GetFaceByPosition ... 8-211
GetFaceIDs ... 8-212
GetFaceIDsOfSheet ... 8-212
GetMatchedObjectName ... 8-213
GetModelBoundingBox ... 8-214

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetModelUnits ... 8-214
GetNumObjects ... 8-215
GetObjectIDByName ... 8-215
GetObjectName ... 8-216
GetObjectNameByEdgeID ... 8-216
GetObjectNameByFaceID ... 8-217
GetObjectNameByID ... 8-217
GetObjectNameByVertexID ... 8-218
GetObjectsByMaterial ... 8-219
GetObjectShapeType ... 8-219
GetObjectsInGroup ... 8-220
GetObjectVolume ... 8-221
GetObjPath [Editor] ... 8-221
GetPartsForUserDefinedModel ... 8-222
GetPoints [3D Modeler Editor] ... 8-222
GetPropEvaluatedValue ... 8-223
GetPropertyValue ... 8-223
GetPropNames [Modeler] ... 8-225
GetPropSIValue ... 8-226
GetPropValue [Modeler] ... 8-227
GetRelativeCoordinateSystems ... 8-228
GetSelections [Model Editor] ... 8-228
GetSubGroupsInGroup ... 8-229
GetUserPosition ... 8-230
GetVertexIDFromNameForFirstOperation ... 8-230
GetVertexIDsFromEdge ... 8-231
GetVertexIDsFromFace ... 8-231
GetVertexIDsFromObject ... 8-232
GetVertexPosition ... 8-232
GetWireBodyNames ... 8-233

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PageSetup ... 8-233
RemoveBadEdges ... 8-235
RemoveBadFaces ... 8-235
RemoveBadVertices ... 8-236
RenamePart ... 8-236
SetPropValue [Modeler] ... 8-237
Setting User-Defined Attributes ... 8-238
SetTopDownViewDirectionForActiveView ... 8-242
SetTopDownViewDirectionForAllViews ... 8-242
UpdatePriorityList ... 8-243
UpgradeVersion ... 8-244
Validate3DComponent ... 8-245
WriteHistoryTreeLayoutForTest ... 8-245
Cable Modeling Commands ... 8-247
AddCableToBundle ... 8-248
CreateCableBundle ... 8-249
CreateCableHarness ... 8-250
CreateClockSource ... 8-252
CreatePWLSource ... 8-253
CreateStraightWireCable ... 8-255
CreateTwistedPairCable ... 8-256
ExportCableLibrary ... 8-257
ImportCableLibrary ... 8-258
RemoveCable ... 8-259
UpdateCableHarness ... 8-259
9 - Output Variable Script Commands ... 9-1
CreateOutputVariable ... 9-1
DeleteOutputVariable ... 9-3
DoesOutputVariableExist ... 9-3
EditOutputVariable ... 9-4

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOutputVariables ... 9-5
GetOutputVariables ... 9-6
GetOutputVariableValue ... 9-7
ImportOutputVariables ... 9-8
SimValueContext ... 9-9
10 - Reporter Editor Script Commands ... 10-1
AddAllEyeMeasurements ... 10-6
AddCartesianLimitLine ... 10-6
AddCartesianLimitLineFromCurve ... 10-7
AddCartesianLimitLineFromEquation ... 10-9
AddCartesianXMarker ... 10-10
AddCartesianYMarker ... 10-11
AddCartesianYMarkerToStack ... 10-11
AddDeltaMarker ... 10-12
AddMarker ... 10-13
AddNote ... 10-14
AddTraceCharacteristics ... 10-15
AddTraces ... 10-17
ApplyReportTemplate ... 10-18
ChangeProperty[ReportSetup] ... 10-19
ClearAllMarkers ... 10-26
ClearAllTraceCharacteristics ... 10-26
CloneReportsFromDatasetSolution ... 10-27
CopyPlotSettings ... 10-28
CopyReportDefinitions ... 10-28
CopyReportsData ... 10-29
CopyTraceDefinitions ... 10-29
CopyTracesData ... 10-30
CreateReport ... 10-31
CreateReportFromTemplate ... 10-43

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateReportOfAllQuantities 10-44
DeleteMarker 10-45
DeleteAllReports 10-45
DeleteReports 10-46
DeleteTraces 10-46
DoesSupportTraceCharacteristics 10-47
EditCartesianXMarker 10-48
EditCartesianYMarker 10-49
EditMarker 10-49
ExportEyeMaskViolation 10-50
ExportImageToFile [Reporter] 10-51
ExportModelImageToFile 10-55
ExportModelMeshToFile 10-58
ExportPlot3DToFile [Reporter] 10-59
ExportReport 10-60
ExportReportDataToFile 10-61
ExportTableToFile 10-61
ExportToFile [Reporter] 10-62
ExportUniformPointsToFile 10-63
GetAllCategories 10-64
GetAllQuantities 10-65
GetAllReportNames 10-66
GetAvailableDisplayTypes 10-67
GetAvailableReportTypes 10-67
GetAvailableSolutions 10-68
GetChildNames [Report Setup] 10-68
GetChildObject [Report Setup] 10-69
GetChildTypes [ReportSetup] 10-69
GetCurvePropServerName 10-70
GetDisplayType 10-71

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDynLinkIntrinsicVariables 10-71
GetDynLinkQtyValueState 10-72
GetDynLinkTraces 10-72
GetDynLinkVariableValues 10-73
GetName 10-74
GetObjPath [Design] 10-74
GetPropertyValue 10-75
GetPropNames [Reporter] 10-76
GetPropValue [Report Setup] 10-77
GetQtyExpressionsForSourceTrace 10-77
GetReportTraceNames 10-78
GetReportSummaryForRegressionTesting 10-79
GetSolutionContexts 10-79
GetSolutionDataPerVariation 10-80
GetDataUnits 10-82
GetDesignVariableNames 10-82
GetDesignVariableUnits 10-83
GetDesignVariableValue 10-84
GetDesignVariationKey 10-85
GetImagDataValues 10-85
GetPerQuantityPrimarySweepValues 10-86
GetRealDataValues 10-87
GetSweepNames 10-88
GetSweepUnits 10-89
GetSweepValues 10-89
IsDataComplex 10-90
IsPerQuantityPrimarySweep 10-91
Release Data 10-92
GroupPlotCurvesByGroupingStrategy 10-93
ImportIntoReport 10-93

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportReportDataIntoReport 10-94
MovePlotCurvesToGroup 10-95
MovePlotCurvesToNewGroup 10-95
OpenWindowForAllReports 10-96
OpenWindowForReports 10-97
PastePlotSettings 10-97
PasteReports 10-98
PasteReportsWithLegacyNames 10-98
PasteTraces 10-99
PasteTracesWithLegacyNames 10-99
RenameReport 10-100
RenameTrace 10-101
ResetPlotSettings 10-101
SavePlotSettingsAsDefault 10-102
SetLinkOutputTraces 10-102
SetPropValue [Report Setup] 10-103
UnGroupPlotCurvesInGroup 10-104
UpdateAllReports 10-105
UpdateReports 10-105
UpdateTraces 10-106
UpdateTracesContextAndSweeps 10-108
11 - Optimetrics Module Script Commands 11-1
CopySetup 11-7
DeleteSetups [Optimetrics] 11-8
DistributedAnalyzeSetup 11-8
EnableSetup 11-9
ExportDXConfigFile 11-10
ExportOptimetricsProfile 11-10
ExportOptimetricsResult 11-11
ExportParametricResults 11-12

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportParametricSetupTable 11-13
ExportRespSurfaceMinMaxTable 11-13
ExportRespSurfaceRefinePoints 11-14
ExportRespSurfaceResponsePoints 11-15
ExportRespSurfaceVerificationPoints 11-15
GenerateVariationData [Parametric] 11-16
GetChildNames [Optimetrics] 11-16
GetChildObject [Optimetrics] 11-17
GetChildTypes [Optimetrics] 11-18
GetName 11-18
GetObjPath [Design] 11-19
GetOptimetricResult 11-20
GetOptimetricsResult 11-20
GetPropNames [Optimetrics] 11-21
GetPropValue [Optimetrics] 11-22
GetSetupNames [Optimetrics] 11-22
GetSetupNamesByType [Optimetrics] 11-23
ImportSetup 11-24
PasteSetup [Optimetrics] 11-25
RenameSetup [Optimetrics] 11-25
SetPropValue [Optimetrics] 11-26
SolveAllSetup 11-27
SolveSetup [Optimetrics] 11-27
ValidateSetup [Parametric] 11-28
General Commands Recognized by the Optimetrics Module 11-29
CopySetup 11-30
DeleteSetups [Optimetrics] 11-31
DistributedAnalyzeSetup 11-32
EnableSetup 11-32
ExportDXConfigFile 11-33

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOptimetricsProfile 11-34
ExportOptimetricsResult 11-34
ExportParametricResults 11-35
ExportParametricSetupTable 11-36
ExportRespSurfaceMinMaxTable 11-36
ExportRespSurfaceRefinePoints 11-37
ExportRespSurfaceResponsePoints 11-38
ExportRespSurfaceVerificationPoints 11-38
ExportDOEResponseCurve 11-39
ExportDOEResponseCurveSlices 11-40
ExportDOEResponseSurface 11-40
ExportDOELocalSensitivity 11-41
ExportDOELocalSensitivityCurve 11-42
GenerateVariationData [Parametric] 11-42
GetChildNames [Optimetrics] 11-43
GetChildObject [Optimetrics] 11-44
GetChildTypes [Optimetrics] 11-44
GetName 11-45
GetObjPath [Design] 11-46
GetOptimetricResult 11-46
GetPropNames [Optimetrics] 11-47
GetPropValue [Optimetrics] 11-47
GetSetupNames [Optimetrics] 11-48
GetSetupNamesByType [Optimetrics] 11-49
ImportSetup 11-49
PasteSetup [Optimetrics] 11-50
RenameSetup [Optimetrics] 11-51
SetPropValue [Optimetrics] 11-51
SolveAllSetup 11-52
SolveSetup [Optimetrics] 11-53

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ValidateSetup [Parametric] 11-54
Parametric Script Commands 11-54
EditSetup [Parametric] 11-55
ExportParametricSetupTable 11-55
GenerateVariationData [Parametric] 11-56
InsertSetup [Parametric] 11-56
ValidateSetup [Parametric] 11-60
Optimization Script Commands 11-61
EditSetup [Optimization] 11-61
InsertSetup [Optimization] 11-65
Sensitivity Script Commands 11-70
EditSetup [Sensitivity] 11-70
Statistical Script Commands 11-74
EditSetup [Statistical] 11-74
InsertSetup [Statistical] 11-75
12 - Solutions Module Script Commands 12-1
DeleteImportData 12-1
DeleteSolutionVariation 12-2
EditQ3DSources (Layout Editor) 12-3
EditSources (Layout Editor) 12-6
ExportEigenmodes 12-10
ExportForHSpice 12-11
ExportNetworkData 12-12
ExportNMFData 12-14
FFTOnReport 12-15
GetAdaptiveFreq 12-16
GetAvailableVariations 12-17
GetExcitationScaling 12-17
GetISolutionVersionID 12-18
GetSolveRangeInfo 12-18

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSourceContexts 12-19
GetSourceData (Layout Editor) 12-19
GetValidISolutionList 12-20
HasFields 12-20
HasMatrixData 12-21
HasMesh 12-21
ImportSolution 12-22
ImportTable 12-23
IsFieldAvailableAt 12-24
ListMatchingVariations 12-25
ListValuesOfVariable 12-25
SetQ3DSATNetChanges (Layout Editor) 12-26
13 - Field Overlays Module Script Commands 13-1
AddMarkerToPlot 13-4
AddNamedExpression 13-5
CalcOp 13-5
CalcStack 13-6
CalculatorRead 13-6
CalculatorWrite 13-7
ChangeProperty 13-8
ChangeGeomSettings 13-11
ClcEval 13-12
ClcMaterial 13-12
ClearAllNamedExpr 13-13
CopyNamedExprToStack 13-13
CreateFieldPlot 13-14
DeleteFieldPlot 13-25
DeleteMarker[Fields Reporter] 13-25
DeleteNamedExpr 13-26
DeleteUneditablePlot 13-26

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DoesNamedExpressionExists 13-27
EditSurfaceMeshSummaryData 13-27
EnterComplex 13-29
EnterComplexVector 13-30
EnterCoord 13-31
EnterEdge 13-31
EnterLine 13-32
EnterOutputVar 13-33
EnterPoint 13-33
EnterQty 13-34
EnterScalar 13-34
EnterScalarFunc 13-35
EnterSurf 13-35
EnterVector 13-36
EnterVectorFunc 13-37
EnterVol 13-37
ExportFieldPlot 13-38
ExportMarkerTable 13-38
ExportOnGrid [Field Overlays] 13-39
ExportPlotImageToFile 13-41
ExportPlotImageWithViewToFile [Reporter] 13-42
ExportSurfaceMeshSummary 13-43
ExportToFile 13-44
GetFieldFolderNames 13-45
GetFieldPlotNames 13-45
GetFieldPlotQuantityName 13-46
GetMeshPlotNames 13-46
GetTopEntryValue 13-47
HideAntennaParameterOverlay 13-48
HidePolarPlot 13-48

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HideRadiatedPlotOverlay 13-49
LoadNamedExpressions 13-49
ModifyFieldPlot 13-50
ReassignFieldPlot 13-52
RenameFieldPlot 13-53
RenamePlotFolder 13-54
SaveFieldsPlots 13-55
SaveNamedExpressions 13-55
SetFieldPlotSettings 13-56
SetPlotFolderSettings 13-58
ShowAntennaParameterOverlay 13-62
UpdateAllFieldsPlots 13-62
UpdateQuantityFieldsPlots 13-63
14 - Fields Calculator Script Commands 14-1
AddNamedExpression 14-2
CalcOp 14-3
CalcRead(deprecated) 14-3
CalculatorRead 14-4
CalcStack 14-5
CalculatorWrite 14-5
ChangeGeomSettings 14-6
ClcEval 14-7
ClcMaterial 14-7
ClcMaterialValue 14-8
ClearAllNamedExpr 14-9
CopyNamedExprToStack 14-9
DeleteNamedExpr 14-10
DoesNamedExpressionExists 14-10
EnterComplex 14-11
EnterComplexVector 14-11

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnterCoord 14-12
EnterEdge 14-13
EnterLine 14-13
EnterOutputVar 14-14
EnterPoint 14-14
EnterQty 14-15
EnterScalar 14-16
EnterScalarFunc 14-16
EnterSurf 14-17
EnterVector 14-17
EnterVectorFunc 14-18
EnterVol 14-18
ExportOnGrid [Fields Calculator] 14-19
ExportToFile [Fields Calculator] 14-21
GetFieldsCalculatorExpressions 14-23
GetTopEntryValue 14-24
LoadNamedExpressions 14-24
SaveNamedExpressions 14-25
15 - Radiation Module Script Commands 15-1
General Commands Recognized by the Radiation Module 15-2
CopyRadFieldSetup 15-2
DeleteFarFieldSetup 15-2
DeleteNearFieldSetup 15-3
DeleteSetup 15-4
EditRadiatedPowerCalculationMethod 15-4
GetChildNames [Radiation] 15-5
GetChildObject [Radiation] 15-5
GetChildTypes [Radiation] 15-6
GetPropNames [Radiation] 15-6
GetPropValue [Radiation] 15-6

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSetupNames 15-7
PasteRadFieldSetup 15-8
RenameSetup [Radiation] 15-8
SetPropValue [Radiation] 15-9
Script Commands for Creating and Modifying Radiation Setups 15-9
AddAntennaOverlay 15-10
AddRadFieldSourceGroup 15-12
EditAntennaOverlay 15-13
EditBoxSetup 15-14
EditFarFieldSphereSetup 15-16
EditInfiniteSphereSetup 15-17
EditLineSetup 15-19
EditNearFieldBoxSetup 15-20
EditNearFieldLineSetup 15-22
EditNearFieldRectangleSetup 15-23
EditNearFieldSphereSetup 15-24
EditRadFieldSourceGroup 15-26
EditRectangleSetup 15-27
EditSphereSetup 15-28
InsertBoxSetup 15-30
InsertFarFieldSphereSetup 15-32
InsertInfiniteSphereSetup 15-33
InsertLineSetup 15-35
InsertNearFieldBoxSetup 15-36
InsertNearFieldLineSetup 15-38
InsertNearFieldRectangleSetup 15-39
InsertNearFieldSphereSetup 15-40
InsertRectangleSetup 15-42
InsertSphereSetup 15-43
Script Commands for Modifying Antenna Array Setups 15-45

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditAntennaArraySetup 15-45
Script Commands for Exporting Antenna Parameters and Max Field Parameters 15-50
ExportElementPatternToFile 15-50
ExportFieldsToFile 15-51
ExportParametersToFile 15-53
ExportRadiationFieldsToFile 15-54
ExportRadiationParametersToFile 15-55
16 - Radiation Setup Manager Module Script Commands 16-1
Add (Infinite Sphere) 16-1
Edit (RadiationSetupMgr for Infinite Sphere) 16-4
Add (Near Field Line) 16-7
Edit (RadiationSetupMgr for Near Field Line) 16-9
Add (Near Field Plane) 16-10
Edit (RadiationSetupMgr for Near Field Plane) 16-13
Add (Near Field Sphere) 16-16
Edit (RadiationSetupMgr for Near Field Sphere) 16-19
17 - User-Defined Document Script Commands 17-1
AddDocument 17-1
DeleteAllDocuments 17-2
DeleteDocument 17-3
EditDocument 17-3
GetDocumentDefinitionNames 17-5
GetDocumentNames 17-5
RenameDocument 17-6
SaveHtmlDocumentAs 17-6
SavePdfDocumentAs 17-7
UpdateAllDocuments 17-8
UpdateDocument 17-8
ViewHtmlDocument 17-9
ViewPdfDocument 17-9

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example Python Script: Defining a Document 17-10
18 - User-Defined Solutions Commands 18-1
CreateUserDefinedSolution 18-1
DeleteUserDefinedSolutions 18-2
EditUserDefinedSolution 18-3
GetUserDefinedSolutionNames 18-4
GetUserDefinedSolutionProperties 18-4
19 - Network Data Explorer Script Commands 19-1
AddDiffPair 19-3
Cascade (SPISim) 19-4
ClearDiffPairs 19-5
Clone 19-6
Close 19-7
Combine (SPISim) 19-8
Deembed (SPISim) 19-9
DeembedBack (SPISim) 19-10
DeembedFront (SPISim) 19-11
DisableDiffPairs 19-12
EnableDiffPairs 19-13
ExportCitiFile 19-13
ExportFullWaveSpice 19-15
ExportMatlab 19-19
ExportNMFData 19-20
ExportNetworkData 19-21
ExportSpreadsheet 19-24
ExportTouchstone 19-26
ExportTouchstone2 19-28
Extract (SPISim) 19-30
GetFrequencies 19-31
GetFrequencyCount 19-32

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetName ... 19-32
GetPortCount ... 19-33
GetPortNumber ... 19-34
GetPostProcSettings ... 19-35
GetSolutionVariation ... 19-36
GetVariation ... 19-36
HasSameData ... 19-37
LoadSolution ... 19-39
Open ... 19-40
Rename (SPISim) ... 19-41
Renormalize (SPISim) ... 19-42
Reorder ... 19-43
Reorder (SPISim) ... 19-44
Reset ... 19-45
SetAllPortImpedances ... 19-46
SetPortDeembedDistance ... 19-47
SetPortImpedance ... 19-47
SetPostProcSettings ... 19-48
Smooth ... 19-49
Stretch (SPISim) ... 19-51
Terminate ... 19-52
20 - CompInstance Script Commands ... (multiple)
Callback Scripting Using CompInstance Object ... (multiple)
CompInstance Functions ... 56
GetComponentName ... 56
GetInstanceID [Component Instance] ... 57
GetInstanceName [Component Instance] ... 57
GetParentDesign ... 58
GetPropHost ... 58
GetPropServerName ... 59

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

21 - Simulation Setup Commands **21-1**
Add (Layout Editor) 21-2
AddMeshOperation 21-14
AddSweep (Layout Editor) 21-19
Analyze 21-21
Analyze (Sweep) 21-22
Delete (Layout Editor) 21-23
DeleteGFD (Setup HFSS 3D Layout) 21-23
DeleteSweep (Layout Editor) 21-24
DynamicMeshOverlays (Layout Editor) 21-24
Edit (Layout Editor) 21-25
EditSweep (Layout Editor) 21-36
ExportToHFSS (Layout Editor) 21-38
ExportToQ3D (Layout Editor) 21-39
GetAllSolutionNames (Layout Editor) 21-39
GetGFDData (Setup HFSS 3D Layout) 21-39
GetMeshOperations 21-40
GetSetupData (Layout Editor) 21-40
LayoutMeshOverlay (Layout Editor) 21-41
Layout3DMeshOverlay (Layout Editor) 21-41
ListVariations (Layout Editor) 21-41
PullExcitationsFromTransient (Layout Editor) 21-41
RefreshMeshOverlays (Layout Editor) 21-44
Rename (Layout Editor) 21-44
RenameSweep (Layout Editor) 21-44
22 - Excitations Commands **22-1**
Add (Excitations) 22-2
AddRefPort (Layout Editor) 22-3
AddRefPortUsingEdges (Layout Editor) 22-6
ConvertPlaneToTrace (Layout Editor) 22-7

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ConvertTraceToPlane (Layout Editor) 22-8
CoupleEdgePorts 22-9
CreateGroundPortComponent (Layout Editor) 22-9
CreateInterfaceGround (Layout Editor) 22-10
CreateInterfacePort (Layout Editor) 22-10
CreateInterfacePortComponent (Layout Editor) 22-10
DecoupleEdgePorts 22-11
Delete[Excitation HFSS 3D Layout] 22-11
Delete [Interface Port] 22-11
Delete[Excitation HFSS 3D Layout] 22-11
DeleteProbePortAndVia (Layout Editor) 22-11
DeleteSource [Interface Source] 22-12
Edit (Layout Editor) 22-12
EditCircuitExcitations (Layout Editor) 22-23
EditCircuitPort [Interface Port] 22-23
EditExcitations (Layout Editor) 22-24
EditHorizontalWavePort (Layout Editor) 22-25
GetAllBoundariesList (Layout Editor) 22-26
GetAllPortsList (Layout Editor) 22-26
Rename [Interface Port] 22-27
RenameSource [Interface Source] 22-27
Rename (Layout Editor) 22-27
RemoveRefPort (Layout Editor) 22-27
SelectInLayout (Layout Editor) 22-28
23 - 2.5D Via Commands **23-1**
ConvertPrimitives (Layout Editor) 23-1
Delete [Via HFSS 3D Layout] 23-2
Edit [Via HFSS 3D Layout] 23-2
MultipleEdit [multiple via HFSS 3D Layout] 23-3
Rename [Via HFSS 3D Layout] 23-3

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SelectInLayout [Via HFSS 3D Layout] 23-3
24 - Layout Scripting **24-1**
Object Identifiers and Script Recording 24-2
Create Primitives 24-3
AddObject ("3D Line") (Layout Editor) 24-3
CreateCircle (Layout Editor) 24-6
CreateLine (Layout Editor) 24-7
AddObject ("Partition Line") (IC Editor) 24-9
CreatePolygon (Layout Editor) 24-12
CreateRectangle (Layout Editor) 24-15
CreateText (Layout Editor) 24-16
Create Voids in Primitives 24-18
CreateCircleVoid (Layout Editor) 24-18
CreateLineVoid (Layout Editor) 24-19
CreatePolygonVoid (Layout Editor) 24-21
CreateRectangleVoid (Layout Editor) 24-22
GetPolygonVoids (Layout Editor) 24-23
Other Creation Methods 24-24
AddPinGroupRefPort (Layout Editor) 24-25
CreateComponent (Layout Editor) 24-26
CreateCurrentSourcePort (Layout Editor) 24-27
CreateHole (Layout Editor) 24-28
CreateMeasure (Layout Editor) 24-30
CreatePin (Layout Editor) 24-32
CreatePinGroup (Layout Editor) 24-34
CreatePinGroupPort (Layout Editor) 24-34
CreatePinGroups (Layout Editor) 24-34
CreateTrace (Layout Editor) 24-35
CreateVia (Layout Editor) 24-38
CreateViaGroups (Layout Editor) 24-38

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateVoltageProbePort (Layout Editor) 24-40
CreateVoltageSourcePort (Layout Editor) 24-41
DeletePinGroup (Layout Editor) 24-42
Object Movement and Modification Methods 24-42
Connect (Layout Editor) 24-43
Copy (Layout Editor) 24-44
Delete (Layout Editor) 24-44
Disconnect (Layout Editor) 24-44
Edit (Layout Editor) 24-45
Expand (Layout Editor) 24-46
ExpandWithPorts (Layout Editor) 24-47
FlipHorizontal (Layout Editor) 24-47
FlipVertical (Layout Editor) 24-48
Move (Layout Editor) 24-48
Paste (Layout Editor) 24-48
Rotate (Layout Editor) 24-48
Activation and Deactivation Methods 24-49
Activate (Layout Editor) 24-49
DeactivateOpen (Layout Editor) 24-50
DeactivateShort (Layout Editor) 24-50
Delete (Layout Editor) 24-51
Layout and Geometry Interrogation Methods 24-51
Layout Interrogation (Layout Editor) 24-51
CreateLineFromPolygon (Layout Editor) 24-52
CreateObjectFromPolygon (Layout Editor) 24-52
FindObjects (Layout Editor) 24-53
FilterObjectList (Layout Editor) 24-54
FindObjectsByPoint (Layout Editor) 24-55
FindObjectsByPolygon (Layout Editor) 24-55
GetBBox (Layout Editor) 24-56

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetCSObjects (Layout Editor) 24-56
GetPolygon (Layout Editor) 24-56
GetPolygonDef (Layout Editor) 24-56
Point (Layout Editor) 24-57
Polygon (Layout Editor) 24-57
Point Object (Layout Editor) 24-57
Set (Layout Editor) 24-58
SetX (Layout Editor) 24-58
GetX (Layout Editor) 24-59
SetY (Layout Editor) 24-59
GetY (Layout Editor) 24-59
SetArc (Layout Editor) 24-59
IsArc (Layout Editor) 24-60
IsEqual (Layout Editor) 24-60
Mag (Layout Editor) 24-60
Distance (Layout Editor) 24-60
Cross (Layout Editor) 24-60
Move (Layout Editor) 24-60
Rotate (Layout Editor) 24-61
Normalize (Layout Editor) 24-61
DistanceFromLine (Layout Editor) 24-61
ClosestPointOnLine (Layout Editor) 24-61
Polygon Object (Layout Editor) 24-62
AddPoint (Layout Editor) 24-64
SetClosed (Layout Editor) 24-64
IsClosed (Layout Editor) 24-65
Move (Layout Editor) 24-65
Rotate (Layout Editor) 24-65
Scale (Layout Editor) 24-65
MirrorX (Layout Editor) 24-66

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPoints (Layout Editor) 24-66
AddHole (Layout Editor) 24-66
HasHoles (Layout Editor) 24-67
GetHoles (Layout Editor) 24-67
HasArcs (Layout Editor) 24-67
HasSelfIntersections (Layout Editor) 24-67
BBoxLL (Layout Editor) 24-68
BBoxUR (Layout Editor) 24-68
IsParametric (Layout Editor) 24-69
IsConvex (Layout Editor) 24-69
IsPoint (Layout Editor) 24-69
IsSegment (Layout Editor) 24-69
IsArc (Layout Editor) 24-69
IsBox (Layout Editor) 24-69
IsCircle (Layout Editor) 24-69
GetBoundingCircleCenter (Layout Editor) 24-70
GetBoundingCircleRadius (Layout Editor) 24-70
Area (Layout Editor) 24-70
PointInPolygon (Layout Editor) 24-70
CircleIntersectsPolygon (Layout Editor) 24-71
GetIntersectionType (Layout Editor) 24-71
GetClosestPoint (Layout Editor) 24-72
GetClosestPoints (Layout Editor) 24-72
Unite (Layout Editor) 24-72
Intersect (Layout Editor) 24-73
Subtract (Layout Editor) 24-73
Xor (Layout Editor) 24-73
Boolean Operations on Primitives 24-74
ConvertPrimitives (Layout Editor) 24-74
ConvertPrimitivestoVias (Layout Editor) 24-75

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Unite (Layout Editor) 24-76
Intersect (Layout Editor) 24-76
SplitRegion (Layout Editor) 24-77
Subtract (Layout Editor) 24-77
ToggleViaPin (Symbol Editor) 24-78
Coordinate System Methods 24-78
CreateCS (Layout Editor) 24-79
ClearRelative (Layout Editor) 24-81
Create3DStructure (Layout Editor) 24-82
GetCSObjects (Layout Editor) 24-83
Group (Layout Editor) 24-83
CreateGroupSelected (Layout Editor) 24-84
PositionRelative (Layout Editor) 24-84
SetCS (Layout Editor) 24-87
Ungroup (Layout Editor) 24-87
IC Editor Script Commands (IC Mode Only) 24-88
Common to General Mode 24-89
ApplyTechnology (IC Editor) 24-94
ChangeActiveLayout (IC Editor) 24-95
Compress Primitives (IC Editor) 24-96
CreateMeshRegion (IC Editor) 24-96
CreateMeshRegionPartitionLine (IC Editor) 24-99
CreateViaGroupsOnLayer (IC Editor) 24-102
ReconstructArcs (IC Editor) 24-104
ReconstructArcsOnLayer (IC Editor) 24-105
RemoveSmallHoles (IC Editor) 24-106
RemoveSmallLines (IC Editor) 24-107
RemoveSmallMetalIslands (IC Editor) 24-107
ResetTechnology (IC Editor) 24-108
SnapPrimitives (IC Editor) 24-108

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SnapPrimitivesOnLayer (IC Editor) 24-110
SnapVias (IC Editor) 24-110
SnapViasOnLayer (IC Editor) 24-112
UnitePrimitivesOnLayer (IC Editor) 24-114
WrapGeometries (IC Editor) 24-114
WrapGeometry (IC Editor) 24-115
Layers Methods 24-116
AddObject (Layout Editor and IC Editor) 24-117
AddLayer (Layout Editor) 24-119
AddStackupLayer (Layout Editor) 24-122
ChangeLayer (Layout Editor) 24-126
ClearLayerMappings (Layout Editor) 24-130
DuplicateAcrossLyrs (Layout Editor) 24-130
ExportGDSII (Layout Editor) 24-131
ExportStackupXML (Layout Editor) 24-133
GetAllLayerNames (Layout Editor) 24-133
GetLayerInfo (Layout Editor) 24-134
GetStackupLayerNames (Layout Editor) 24-136
ImportStackupXML (Layout Editor) 24-136
RemoveLayer (Layout Editor) 24-137
General Mode 24-137
IC Mode (IC Editor) 24-137
SetLayerMapping (Layout Editor) 24-138
SyncRigidFlexSandbox (Layout Editor) 24-139
Net and NetClass Methods 24-140
CreateDifferentialPair (Layout Editor) 24-141
CreateNetClass 24-142
DeleteNets (Layout Editor) 24-142
DelNetClass 24-143
GetNetClasses 24-143

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNetClassNets ... 24-144
GetNetConnections (Layout Editor) ... 24-144
GetNets (Layout Editor) ... 24-145
HighlightNet (Layout Editor) ... 24-145
ModifyDifferentialPair (Layout Editor) ... 24-146
ModifyNetClass ... 24-147
ResetDifferentialpairs (Layout Editor) ... 24-148
SelectNetConnected (Layout Editor) ... 24-150
SelectPhysicallyConnected (Layout Editor) ... 24-151
SetNetColor (Layout Editor) ... 24-152
SetNetVisible (Layout Editor) ... 24-153
ToggleNetHighlight (Layout Editor) ... 24-155
Port Methods ... 24-156
AddCircuitRefPort (Layout Editor) ... 24-157
AddPortsToAllNets (Layout Editor) ... 24-159
AddPortsToNet (Layout Editor) ... 24-159
AddRefPort (Layout Editor) ... 24-160
AlignPorts (Layout Editor) ... 24-163
AssignCircuitRefPort (Layout Editor) ... 24-164
ClearRefPort (Layout Editor) ... 24-165
CreateCircuitPort (Layout Editor) ... 24-166
CreateEdgePort (Layout Editor) ... 24-167
CreateNPort (Layout Editor) ... 24-169
CreatePortInstancePorts (Layout Editor) ... 24-171
CreatePortsOnComponents (Layout Editor) ... 24-171
CreatePortsOnComponentsByNet (Layout Editor) ... 24-172
GetPortInfo (Layout Editor) ... 24-173
RemovePort (Symbol Editor) ... 24-176
RemovePortsFromAllNets (Layout Editor) ... 24-177
RemovePortsFromNet (Layout Editor) ... 24-177

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemovePortsOnComponents (Layout Editor) 24-178
SetHfssExtentsVisible (Layout Editor) 24-179
SetHfssPortsVisible (Layout Editor) 24-179
Miscellaneous Methods 24-180
AlignObjects (Layout Editor) 24-182
ChangeOptions (Layout Editor) 24-184
ChangeProperty (Schematic Editor and Layout Editor) 24-189
Property Array Structure 24-190
ClipPlane (Layout Editor) 24-195
CopyToHFSS 3D Layout (Layout Editor) 24-196
CutOutSubDesign (Layout Editor) 24-196
Defeature Objects (Layout Editor) 24-200
DissolveComponents (Layout Editor) 24-201
Duplicate (Layout Editor) 24-202
Edit3DComponentDefinition 24-202
EditComponent (Layout Editor) 24-203
EditPinGroup (Layout Editor) 24-205
EnableComponents (Layout Editor) 24-206
EraseMeasurements (Layout Editor) 24-207
ExportDXF (Layout Editor) 24-207
ExportGDSII (Layout Editor) 24-209
ExportGerber (Layout Editor) 24-211
ExportNCDrill (Layout Editor) 24-212
GetActiveUnits (Layout Editor) 24-214
GetComponentInfo (Layout Editor) 24-215
GetCompInstanceFromRefDes (Layout Editor) 24-216
GetComponentPins (Layout Editor) 24-216
GetComponentPinInfo (Layout Editor) 24-217
GetEditorName (Layout Editor) 24-218
GetMaterialList (Layout Editor) 24-218

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProperties (Layout Editor) ... 24-218
GetPropertyValue (Layout Editor) ... 24-219
GetSelections (Layout Editor) ... 24-219
GetSelectionsInHierarchy (Layout Editor) ... 24-219
Heal (Layout Editor) ... 24-220
PageSetup (Layout Editor) ... 24-222
PushExcitations ... 24-223
Select (Layout Editor) ... 24-225
SelectAll (Layout Editor) ... 24-225
SetActiveUnits (Layout Editor) ... 24-226
SetPropertyValue (Layout Editor) ... 24-226
StitchLines (Layout Editor) ... 24-227
UnselectAll (Layout Editor) ... 24-228
ZoomToFit (Layout Editor) ... 24-228

25 - Definition Manager Script Commands ... 25-1

Add [component manager] ... 25-3
AddDataset ... 25-6
AddDefinitionFromBlock ... 25-8
AddMaterial ... 25-10
Add [padstack manager] ... 25-14
Add [symbol manager] ... 25-20
AreMaterialPropertiesEqual ... 25-28
AreSurfaceMaterialPropertiesEqual ... 25-29
ChangeProperty (Symbol Editor) ... 25-30
CloneMaterial ... 25-40
DeleteDataset ... 25-40
DoesMaterialExist ... 25-41
Edit [component manager] ... 25-42
EditDataset ... 25-54
EditMaterial ... 25-56

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit [padstack manager] ... 25-63
ExportDataset ... 25-68
Export [footprint manager] ... 25-68
ExportMaterial ... 25-69
Export [padstack manager] ... 25-70
ExportScript ... 25-71
Export [symbol manager] ... 25-71
GetProjectMaterialNames ... 25-72
GetPropertyValue ... 25-73
ImportDataset ... 25-74
Note About File Types: ... 25-75
Remove [component manager] ... 25-76
Remove [footprint manager] ... 25-77
RemoveMaterial ... 25-78
Remove [padstack manager] ... 25-79
RemoveScript ... 25-79
Remove [symbol manager] ... 25-80
RemoveUnusedDefinitions ... 25-81
SetPropertyValue ... 25-82
UpdateDefFromBlock ... 25-83
UpdateDefFromBlockEx ... 25-85
UpdateDefinitions ... 25-88
Component Manager Script Commands ... 25-89
Add [component manager] ... 25-90
AddNPortData [component manager] ... 25-92
Add (for Solver On Demand Model) ... 25-97
ClearSolutionCache [component manager] ... 25-104
Edit [component manager] ... 25-105
EditSolverOnDemandModel ... 25-117
EditWithComps [component manager] ... 25-118

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Export [component manager] ... 25-124
GetNames [component manager] ... 25-125
GetNPortData [component manager] ... 25-126
GetSolverOnDemandData ... 25-126
GetSolverOnDemandModelList ... 25-127
IsUsed [component manager] ... 25-127
Remove [component manager] ... 25-128
RemoveSolverOnDemandModel ... 25-129
RemoveUnused [component manager] ... 25-129
Update Dynamic Link [component manager] ... 25-130
Footprint Manager Script Commands ... 25-130
Add [footprint manager] ... 25-131
Edit [footprint manager] ... 25-143
EditWithComps [footprint manager] ... 25-143
Export [footprint manager] ... 25-153
GetNames [footprint manager] ... 25-154
IsUsed [footprint manager] ... 25-154
Remove [footprint manager] ... 25-155
RemoveUnused [footprint manager] ... 25-156
Material Manager Script Commands ... 25-156
GetNames [material manager] ... 25-157
GetProperties [material manager] ... 25-157
IsUsed [material manager] ... 25-158
RemoveUnused [material manager] ... 25-159
Model Manager Script Commands ... 25-159
Add [model manager] ... 25-160
ConvertToDynamic ... 25-166
ConvertToParametric ... 25-166
Edit [deprecated] ... 25-166
EditWithComps [model manager] ... 25-166

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Export [model manager] 25-173
GetNames [model manager] 25-174
IsUsed [model manager] 25-174
Remove [model manager] 25-175
RemoveUnused [model manager] 25-175
Network Data Explorer Manager Script Commands 25-176
ExportFullWaveSpice 25-177
ExportNetworkData 25-180
ExportNMFData 25-182
Padstack Manager Script Commands 25-183
Add [padstack manager] 25-184
Edit [padstack manager] 25-190
EditWithComps [padstack manager] 25-195
Export [padstack manager] 25-201
GetNames [padstack manager] 25-202
IsUsed [padstack manager] 25-202
Remove [padstack manager] 25-203
RemoveUnused [padstack manager] 25-203
Script and Library Scripts 25-204
AddScript 25-204
EditScript 25-205
ExportScript 25-206
ModifyLibraries 25-207
RemoveScript 25-208
Symbol Manager Script Commands 25-209
Add [symbol manager] 25-209
BringToFront [symbol manager] 25-218
Edit [deprecated] 25-218
EditSymbolAndUpdateComps [symbol manager] 25-218
Export [symbol manager] 25-227

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNames [symbol manager] 25-228
IsUsed [symbol manager] 25-228
Remove [symbol manager] 25-229
RemoveUnused [symbol manager] 25-229
Technology Manager Script Commands 25-230
Add (Technology Manager) 25-231
GetNames (Technology Manager) 25-232
IsUsed (Technology Manager) 25-233
Remove (Technology Manager) 25-233
26 - Definition Editor Script Commands 26-1
AddCircuitRefPort (Layout Editor) 26-7
AddLayer (Layout Editor) 26-9
AddPortsToAllNets (Layout Editor) 26-12
AddPortsToNet (Layout Editor) 26-13
AddRefPort (Layout Editor) 26-14
AlignObjects (Layout Editor) 26-17
AlignPorts (Layout Editor) 26-19
AssignCircuitRefPort (Layout Editor) 26-20
AssignRefPort (Layout Editor and IC Mode) 26-21
ChangeProperty 26-24
ClearLayerMappings (Layout Editor) 26-27
ClearRefPort (Layout Editor) 26-28
ClearRelative (Layout Editor) 26-29
Connect (Layout Editor) 26-29
ConvertPlaneToTrace (Layout Editor) 26-30
ConvertTraceToPlane (Layout Editor) 26-31
Copy (Layout Editor) 26-31
CopyToHFSS 3D Layout (Layout Editor) 26-32
Create3DStructure (Layout Editor) 26-32
CreateCS (Layout Editor) 26-33

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateCircle (Layout Editor) 26-36
CreateCircleVoid (Layout Editor) 26-37
CreateCircuitPort (Layout Editor) 26-39
CreateCircuitPortsOnComponents (Layout Editor) 26-40
CreateComponent (Layout Editor) 26-41
CreateEdgePort (Layout Editor) 26-43
CreateGroundPortComponent (Layout Editor) 26-45
CreateGroupSelected (Layout Editor) 26-46
CreateHole (Layout Editor) 26-47
CreateInterfaceGround (Layout Editor) 26-48
CreateInterfacePort (Layout Editor) 26-49
CreateInterfacePortComponent (Layout Editor) 26-49
CreateLine (Layout Editor) 26-49
CreateLineFromPolygon (Layout Editor) 26-51
CreateLineVoid (Layout Editor) 26-51
CreateMeasure (Layout Editor) 26-53
CreateNportCircuitElements 26-55
CreateNetClass 26-57
CreateObjectFromPolygon (Layout Editor) 26-57
CreatePin (Layout Editor) 26-58
CreatePinGroup (Layout Editor) 26-59
CreatePinGroupPort (Layout Editor) 26-59
CreatePinGroupPortsOnComponents (Layout Editor) 26-59
CreatePolygon (Layout Editor) 26-61
CreatePolygonVoid (Layout Editor) 26-63
CreatePortInstancePorts (Layout Editor) 26-65
CreatePortsOnComponents (Layout Editor) 26-65
CreateRectangle (Layout Editor) 26-66
CreateRectangleVoid (Layout Editor) 26-67
CreateText (Layout Editor) 26-69

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateTrace (Layout Editor) 26-70
CreateVia (Layout Editor) 26-73
CutOutSubDesign (Layout Editor) 26-74
DeactivateOpen (Layout Editor) 26-77
DeactivateShort (Layout Editor) 26-78
Defeature Objects (Layout Editor) 26-78
DelNetClass 26-79
Delete (Layout Editor) 26-79
DeleteNets (Layout Editor) 26-79
DeletePinGroup (Layout Editor) 26-80
Disconnect (Layout Editor) 26-80
Duplicate (Layout Editor) 26-81
DuplicateAcrossLyrs (Layout Editor) 26-81
Edit (Layout Editor) 26-82
Edit3DComponentDefinition 26-83
EraseMeasurements (Layout Editor) 26-84
Expand (Layout Editor) 26-84
ExpandWithPorts (Layout Editor) 26-85
ExportDXF (Layout Editor) 26-86
ExportGDSII (Layout Editor) 26-87
ExportGerber (Layout Editor) 26-89
ExportNCDrill (Layout Editor) 26-91
ExportToHFSS (Layout Editor) 26-93
ExportToQ3D (Layout Editor) 26-93
FilterObjectList (Layout Editor) 26-93
FindObjectsByPoint (Layout Editor) 26-94
FindObjectsByPolygon (Layout Editor) 26-95
FlipHorizontal (Layout Editor) 26-95
FlipVertical (Layout Editor) 26-95
GeometryCheckAndAutofix 26-95

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAllLayerNames (Layout Editor) 26-97
GetBBox (Layout Editor) 26-97
GetCSObjects (Layout Editor) 26-97
GetCompInstanceFromRefDes (Layout Editor) 26-97
GetComponentInfo (Layout Editor) 26-98
GetComponentPinInfo (Layout Editor) 26-99
GetComponentPins (Layout Editor) 26-99
GetEditorName (Layout Editor) 26-100
GetMaterialList (Layout Editor) 26-101
GetNetClassNets 26-101
GetNetClasses 26-101
GetNetConnections (Layout Editor) 26-102
GetPolygon (Layout Editor) 26-103
GetPolygonDef (Layout Editor) 26-103
GetPortInfo (Layout Editor) 26-103
GetProperties (Layout Editor) 26-106
GetPropertyValue (Layout Editor) 26-106
GetSelections [Design] 26-106
Group (Layout Editor) 26-106
Heal (Layout Editor) 26-107
HighlightNet (Layout Editor) 26-109
ImportDXF [Modeler] 26-110
ImportGDSII 26-113
Intersect (Layout Editor) 26-113
ModifyNetClass 26-114
Move (Layout Editor) 26-114
PageSetup (Layout Editor) 26-114
Point (Layout Editor) 26-115
Polygon (Layout Editor) 26-115
PositionRelative (Layout Editor) 26-115

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PushExcitations 26-118
RemoveLayer (Layout Editor) 26-120
RemovePortsFromAllNets (Layout Editor) 26-121
RemovePortsFromNet (Layout Editor) 26-121
RemovePortsOnComponents (Layout Editor) 26-122
Rotate (Layout Editor) 26-123
SelectAll (Layout Editor) 26-123
SetCS (Layout Editor) 26-124
SetLayerMapping (Layout Editor) 26-124
SetNetVisible (Layout Editor) 26-125
StitchLines (Layout Editor) 26-126
Subtract (Layout Editor) 26-127
ToggleViaPin (Footprint Editor) 26-127
Ungroup (Layout Editor) 26-128
Unite (Layout Editor) 26-128
UnselectAll (Layout Editor) 26-129
ZoomToFit (Layout Editor) 26-129
Symbol Editor Scripts 26-130
AddLevel (Symbol Editor) 26-131
AlignHorizontal (Symbol Editor) 26-131
AlignVertical (Symbol Editor) 26-132
BringToFront (Symbol Editor) 26-133
ChangeProperty (Symbol Editor) 26-134
CloseEditor (Symbol Editor) 26-144
Copy (Symbol Editor) 26-144
CreateArc (Symbol Editor) 26-145
CreateCircle (Symbol Editor) 26-146
CreateImage (Symbol Editor) 26-146
CreateLine (Symbol Editor) 26-147
CreatePin (Symbol Editor) 26-148

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreatePolygon (Symbol Editor) 26-148
CreateRectangle (Symbol Editor) 26-149
CreateText (Symbol Editor) 26-149
Cut (Symbol Editor) 26-150
Delete (Symbol Editor) 26-150
DisplayPinNames (Symbol Editor) 26-151
ExportFile (Symbol Editor) 26-151
FlipHorizontal (Symbol Editor) 26-152
FlipVertical (Symbol Editor) 26-152
GetProperties (Symbol Editor) 26-153
GetPropertyValue (Symbol Editor) 26-153
GetVisibleLevels (Symbol Editor) 26-153
GridSetup (Symbol Editor) 26-154
Move (Symbol Editor) 26-155
Pan (Symbol Editor) 26-156
Paste (Symbol Editor) 26-156
Redo (Symbol Editor) 26-157
RemoveLevels (Symbol Editor) 26-157
RemovePort (Symbol Editor) 26-158
Rotate (Symbol Editor) 26-158
Save (Symbol Editor) 26-159
SelectAll (Symbol Editor) 26-159
SendToBack (Symbol Editor) 26-160
SetActivelLevel (Symbol Editor) 26-160
SetInitialLevels (Symbol Editor) 26-161
SetPropertyValue (Symbol Editor) 26-161
SetVisibleLevels (Symbol Editor) 26-162
ToggleLevel (Symbol Editor) 26-163
ToggleViaPin (Symbol Editor) 26-163
Undo (Symbol Editor) 26-164

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ZoomArea (Symbol Editor) 26-164
ZoomIn (Symbol Editor) 26-165
ZoomOut (Symbol Editor) 26-166
ZoomPrevious (Symbol Editor) 26-166
ZoomToFit (Symbol Editor) 26-167
Footprint Editor Scripts 26-167
AddLayer (Footprint Editor) 26-169
AddStackupLayer (Footprint Editor) 26-172
ChangeLayer (Footprint Editor) 26-175
ChangeLayers (Footprint Editor) 26-178
ChangeOptions (Footprint Editor) 26-184
CloseEditor (Footprint Editor) 26-189
CreateCircle (Footprint Editor) 26-189
CreateCircleVoid (Footprint Editor) 26-191
CreateEdgePort (Footprint Editor) 26-192
CreateLine (Layout Editor) 26-193
CreateLineVoid (Footprint Editor) 26-195
CreatePolygonVoid (Footprint Editor) 26-195
CreatePin (Footprint Editor) 26-196
CreatePolygon (Footprint Editor) 26-196
CreateRectangle (Footprint Editor) 26-199
CreateText (Footprint Editor) 26-199
CreateVia (Footprint Editor) 26-199
Duplicate (Footprint Editor) 26-199
Edit (Footprint Editor) 26-200
EraseMeasurements (Footprint Editor) 26-200
FlipHorizontal (Footprint Editor) 26-200
FlipVertical (Footprint Editor) 26-201
GetAllLayerNames (Footprint Editor) 26-201
GetLayerInfo (Footprint Editor) 26-201

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProperties (Footprint Editor) ... 26-203
GetStackupLayerNames (Footprint Editor) ... 26-204
Intersect (Footprint Editor) ... 26-204
Move (Footprint Editor) ... 26-204
PageSetup (Footprint Editor) ... 26-205
RemoveLayer (Footprint Editor) ... 26-205
RemovePort (Footprint Editor) ... 26-205
Rotate [Footprint Editor] ... 26-206
Save (Footprint Editor) ... 26-206
SetActiveDefinitionEditor (Footprint Editor) ... 26-207
SetPropertyValue (Footprint Editor) ... 26-208
Subtract (Footprint Editor) ... 26-208
ToggleViaPin (Footprint Editor) ... 26-209
Unite (Footprint Editor) ... 26-209
ZoomToFit (Footprint Editor) ... 26-210
27 - Core Global Script Context Commands ... 27-1
AddErrorMessage ... 27-1
AddFatalMessage ... 27-1
AddInfoMessage ... 27-2
AddWarningMessage ... 27-2
LogDebug ... 27-3
LogError ... 27-3
28 - Example Scripts ... 28-1
IronPython Example Scripts ... 28-1
BH Coordinates Python Script ... 28-1
Script Contents ... 28-2
Posco_BH_Curve.tab Contents ... 28-3
Equation Based Curve Python Script ... 28-6
Index ... Index-1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

1 - Introduction to Scripting

Using scripts is a fast, effective way to accomplish tasks you want to repeat. When you execute a script, the commands in the script are performed in the order in which they appear.

Electronics Desktop can record scripts in Python, and can run external scripts written in Python. Additionally, it contains a Python command shell for executing scripts.

When running Ansys Electronics Desktop from the command line, scripts can be written in any language that provides Microsoft COM methods.

The following sections contain more information about scripting:

- Scripting Help Conventions – explains the layout of the scripting help.
- Introduction to IronPython – provides a broad overview of IronPython.
- Introduction to C-Python – provides guidance on using C-Python for Ansys Electronics Desktop scripts.
- Ansys Electronics Desktop Scripting – details instructions and tips for running, recording, and working with scripts in Electronics Desktop.
- PyAEDT (Beta) – a Python library that interacts directly with the AEDT API to make scripting simpler for the end user.

Scripting Help Conventions

The majority of this guide lists individual script commands using the following format.

[ScriptName]

[Description of script use.]

UI Access	[UI commands corresponding to the script command, if any.]
Parameters	[List of arguments taken by the script command, if any. Includes argument types and brief descriptions.]
Return Value	[The script's return value, if any.]

Python Syntax	[Correct syntax for the command in Python. Arguments are enclosed in angle brackets (<>).]
Python Example	`[Sample script]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Variable Types

The following data types are used throughout the help:

- **<string>** – use within quotation marks.
- **<bool>** – boolean value; should be set to either True or False.
- **<int>** – an integer. For example, 1.
- **<double>** – a double precision value. For example, 1.2.
- **<array>** – a list contained in square brackets.
- **<value>** – can be an integer, string, or other variable, depending on context.

Introduction to IronPython

IronPython is an implementation of the Python programming language targeting the .NET runtime. What this means in practical terms is that IronPython uses the Python programming language syntax and standard python libraries and can additionally use .NET classes and objects to give one the best of both worlds. This usage of .NET classes is fairly seamless in that a class defined in a .NET assembly can be used as a base class of a python class.

Scope

Functioning as a tutorial on Python or IronPython is way out of the scope of this document. There are several excellent resources online that do a very good job in that regard. This document only attempts to provide a limited introduction to IronPython as used to script Ansys EM products.

This document is also not a tutorial on the scripting of Ansys EM products. It complements the existing scripting guide (available from a product's Help menu) and provides a pythonic interpretation of that information.

Python Compatibility

The version of IronPython in use is **2.7** and built on the .NET framework version 4.0: this version targets **Python 2.7** language compatibility. While most python files will execute under IronPython with no changes, python libraries that make use of extensions written in the C programming language (NumPy or SciPy for instance), are not expected to work under IronPython. In such cases, it might be possible to locate .NET implementation of such libraries or explore the use of IronClad.

(http://code.google.com/p/ironclad/).

Advantages of IronPython

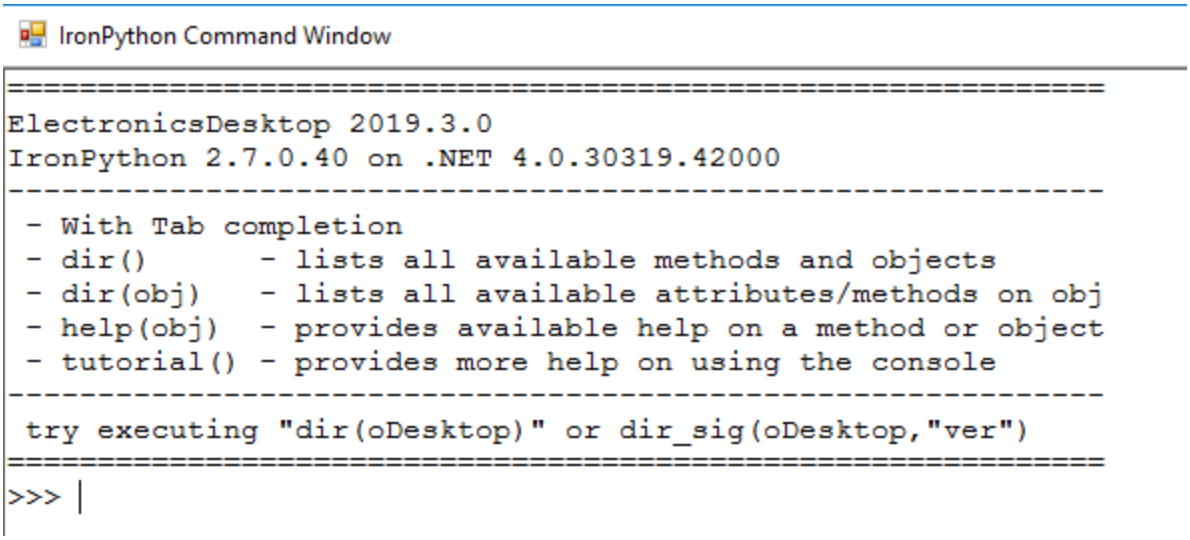
The advantages that IronPython use provides are significant:

- Python has a large eco-system with plenty of supporting libraries, Visual IDEs and debuggers. It is actively developed and enhanced.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- IronPython, in addition, has access to the entire .NET eco system. This allows us, for instance, to create a modern GUI using the **System.Windows.Forms** assembly from IronPython code and call any other .NET assembly for that matter.
- The use of IronPython's technologies enables the ability to interactively script Desktop (feature in development).
- The Python syntax of dictionaries is somewhat easier to read and write when supplying arguments to the scripting methods.

This document describes IronPython briefly and then goes on to describe the desktop provided IronPython scripting console and scripting with IronPython. You can open an IronPython Command Window by clicking **Tools** > **Open Command Window**.



Scripting Using Iron Python

Standalone IronPython and Desktop IronPython

IronPython Examples

Creating User Defined Primitives and User Defined Models in Python Scripts

Scripting Using Iron Python

The following topics detail scripting using Iron Python:

IronPython Script Execution Environment

Scripting with IronPython

Standalone IronPython and Desktop IronPython

Introduction to IronPython

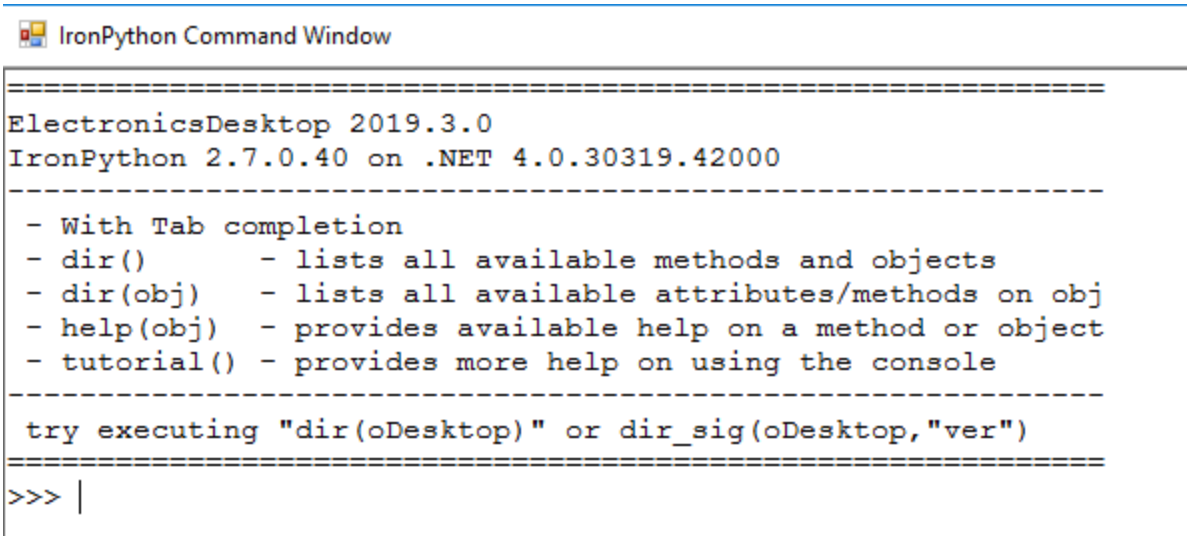
Appendix: IronPython Samples

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IronPython Script Execution Environment

Scripts written in IronPython are executed by desktop in four different ways:

- **Tools** > **Open Command Window**, to open the **IronPython Command Window:**



- **Tools** > **Run Script** menu item, select "IronPython" from the file type drop-down list.
- Launch the product with a script argument.
  - HFSS -runscript someScript.py to keep HFSS GUI open after completing script execution.
  - HFSS -features=beta -ng -runscriptandexit someScript.py to run HFSS in a non-graphical mode and exit after script completion. Note that this is a beta feature supported for the HFSS and 3D Layout design types.
- Register an IronPython script as an external tool using the **Tools** > **External Tools** menu item.

When desktop executes a script, it does so in an execution environment setup with predefined variables and functions. These predefined variables and functions are how the script communicates with the desktop, and they come in four flavors addressed in the following subtopics:

Script Argument for IronPython

Script Argument for IronPython

When scripts are launched using the **Tools** > **Run Script** menu item, the dialog box that pops up allows the user to specify arguments.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

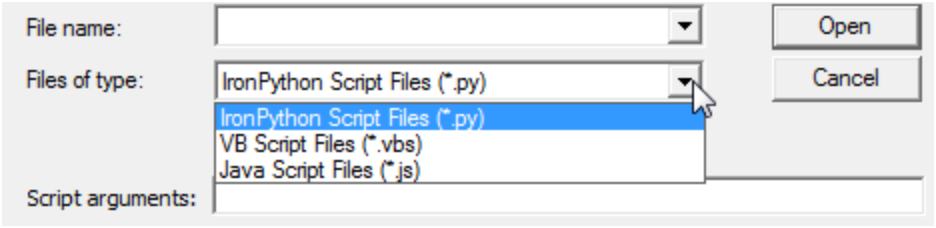


Figure 1: Run Script dialog and script arguments

Any argument specified here is communicated to the script being executed as the predefined variable **ScriptArgument**.

Related Topics

IronPython Script Execution Environment

Scripting using Embedded VBScript or JavaScript

While Ansys Electronics Desktop no longer supports VBScript or JavaScript, users may have a significant collection of VBScript or JavaScript assets. These existing script files can be executed via Python. Various **Run<*>Command** methods have been designed for this purpose.

For example, a user can create a parameterized cone in HFSS by executing the following Python script from the **Tools** > **Run Script** menu:

```
# assign the VBScript snippet obtained from a script recording from
HFSS to

# coneScript and replace the BottomRadius recorded value with botRa-
dius

coneScript = """Dim oAnsoftApp

Dim oDesktop

Dim oProject

Dim oDesign

Dim oEditor

Dim oModule

Set oAnsoftApp = CreateObject("Ansoft.ElectronicsDesktop")

Set oDesktop = oAnsoftApp.GetAppDesktop()

oDesktop.RestoreWindow

Set oProject = oDesktop.GetActiveProject()

oProject.InsertDesign "HFSS", "HFSSPyTestDesign", "DrivenModal", ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Set oDesign = oProject.SetActiveDesign("HFSSPyTestDesign")

Set oEditor = oDesign.SetActiveEditor("3D Modeler")

oEditor.CreateCone Array("NAME:ConeParameters", _

 "XCenter:=", "0mm", "YCenter:=", "0mm", "ZCenter:=", "0mm",_

 "WhichAxis:=", "Z", "Height:=", "2mm", _

 "BottomRadius:=", "3mm", _

 "TopRadius:=", "0mm"), Array("NAME:Attributes", "Name:=", _

 "Cone1", "Flags:=", "", "Color:=", "(132 132 193)", "Trans-
parency:=", 0, _

 "PartCoordinateSystem:=", "Global", "UDMId:=", "", "Mater-
ialValue:=", _

 "" & Chr(34) & "vacuum" & Chr(34) & "", "SolveInside:=", _

 true)

"""


SetScriptingLanguageToVBScript()

RunScriptCommand(coneScript, "")
```

This hybrid approach is useful when you have existing VBScript commands that you want to reuse or when you want to quickly parameterize a recorded sample, but one significant limitation of this approach is the inability to capture return values from VBScript or JavaScript calls that do return something. Full two-way communication with the product requires the use of pure Python to directly invoke the script objects.

Related Topics

IronPython Script Execution Environment

Scripting with Python

Access to application scripting objects is provided via the predefined **oDesktop** object.

Note the following:

- Any argument is supplied via the built in **ScriptArgument** variable.
- The **oDesktop** object is always available.
- Method calls have to adhere to the rule of ensuring trailing parentheses irrespective of whether the function returns anything or has any arguments.
- Any compound/block arguments should be translated to the appropriate Python array or dictionary syntax.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Related Topics

IronPython Script Execution Environment

Standalone IronPython and Desktop IronPython

Standalone IronPython

In general, it is easier to run a script directly from Electronics Desktop. Standalone IronPython does not implement all the functionality available when a script is run from Electronics Desktop. It only implements full support for COM functions.

Running Standalone IronPython

Standalone IronPython uses COM to get the handle to the AnsysEDT app. To run standalone IronPython, you'll need to call the IronPython interpreter ipy64.exe.

It is located in:

```
\\<AnsysEDTInstallationPath>\common\IronPython\ipy64.exe
```

For example, to run myScript.py, type the following in the command line:

```
"C:\Program Files\ANSYS Inc\v261\AnsysEM\common\IronPython\ipy64.exe"
"<filePath>\myScript.py"
```

You can set the interpreter to be the default program when double-clicking the .py script. You can use any recorded script as the basis for a standalone script and simply add an installation-internal path to the python module search path (as shown below) and end the script with a new shutdown call.

Using a Recorded Script

A python script recorded in AnsysEDT already has the required lines to be run as a standalone, except for the first two lines (path settings) and the final `Shutdown()` call. See the example script below.

Creating an External Script

When creating a script outside of Electronics Desktop, the following lines should be included at the beginning of your script:

- `import sys`

  # Imports the sys module containing system-specific functions native to IronPython.

- `sys.path.append("<InstallationPath>")`

  # Adds the Electronics Desktop installation path to the list of directories Python searches for modules and files.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- `sys.path.append("<InstallationPath>/PythonFiles/DesktopPlugin")`

  # Adds the PythonFiles/DesktopPlugin subfolder to the list of directories Python searches for modules and files.

- `import ScriptEnv`

  # This imports ScriptEnv.py from the installation path specified above. ScriptEnv.py performs an operating system check and defines functions used in Electronics Desktop scripts. See the annotations in the ScriptEnv.py file for more information.

- `ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")`

  or `ScriptEnv.InitializeNew(NonGraphical=True)`

  # `Initialize` and `InitializeNew` are functions within ScriptEnv.py. The first option launches Electronics Desktop. The second allows you to run a script without launching Electronics Desktop. See the annotations in the ScriptEnv.py file for more information.

You must end the script with:

- `ScriptEnv.Shutdown()`

  # This stops ScriptEnv.py. If you are running multiple scripts, include this only at the end of the last script.

Example Script

```
import sys

sys.path.append(r"C:\Program Files\ANSYS Inc\v261\AnsysEM")

sys.path.append(r"C:\Program Files\ANSYS Inc\v261\An-
sysEM\PythonFiles\DesktopPlugin")


import ScriptEnv

ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")

oDesktop.RestoreWindow()

oProject = oDesktop.NewProject()

oProject.InsertDesign("HFSS", "HFSSDesign1", "DrivenModal", "")

oDesign = oProject.SetActiveDesign("HFSSDesign1")

oEditor = oDesign.SetActiveEditor("3D Modeler")

oEditor.CreateRectangle(

[

  "NAME:RectangleParameters",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "IsCovered:= ", True,
    "XStart:= ", "-0.2mm",
    "YStart:= ", "-3mm",
    "ZStart:= ", "0mm",
    "Width:= ", "0.8mm",
    "Height:= ", "1.2mm",
    "WhichAxis:= ", "Z"
],
[
    "NAME:Attributes",
    "Name:= ", "Rectangle1",
    "Flags:= ", "",
    "Color:= ", "(132 132 193)",
    "Transparency:= ", 0,
    "PartCoordinateSystem:=", "Global",
    "UDMId:= ", "",
    "MaterialValue:= ", "\"vacuum\"",
    "SolveInside:= ", True
])
oDesign.SetDesignSettings(['NAME:Design Settings Data', 'Allow Mater-
ial Override:=', True, 'Calculate Lossy Dielectrics:=', True])
oEditor.SetModelUnits(['NAME:Units Parameter', 'Units:=', 'mil', 'Res-
cale:=' , False ])
ScriptEnv.Shutdown()
```

IronPython Samples

> **Important:**
>
> VBScript is no longer supported in Ansys Electronics Desktop. It is referenced here only to aid users in translating existing VBScript scripts to Python.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Change property

The following snippets show how a change property command (in this case, to change the color of a cone) looks in VBScript and its two possible IronPython variants.

```
oEditor.ChangeProperty Array("NAME:AllTabs", Array("NAME:Geo-
metry3DAttributeTab",_
  Array("NAME:PropServers", "Cone1"), _
  Array("NAME:ChangedProps", _
  Array("NAME:Color", "R:=", 255, "G:=", 255, "B:=", 0))))
```

Sample Script: ChangeProperty command to change color of a cone

```
oEditor.ChangeProperty(
["NAME:AllTabs",
   ["NAME:Geometry3DAttributeTab",
   ["NAME:PropServers", "Cone1"],
   ["NAME:ChangedProps",
   ["NAME:Color", "R:=", 0, "G:=", 0, "B:=", 64]
   ]
 ]
    ])
```

Sample Script: ChangeProperty command to change color of cone using Python arrays

Any time there are named arrays composed purely of key-value pairs, they can always be represented using a Python dictionary, irrespective of the nesting of said named array.

```
oEditor.ChangeProperty(
   ["NAME:AllTabs",
   ["NAME:Geometry3DAttributeTab",
     ["NAME:PropServers", "Cone1"],
     ["NAME:ChangedProps",
     {
    "NAME":"Color",
    "R" : 0,
    "G" : 64,
    "B" : 0
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   }]]
 ])
```

Sample Script: ChangeProperty command to change the color of a cone using Python arrays and dictionaries

Create a Cone using IronPython

Most scripting tasks using IronPython are expected to be formatted as the following example. One starts with the predefined **oDesktop** object and drills down to the design, editors, modules etc and issues any required commands on the object while formatting the script command arguments in natural python syntax.

```
oProject = oDesktop.GetActiveProject()
oDesign = oProject.InsertDesign("HFSS","Random","DrivenModal","")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCone(
 {
   "NAME" : "ConeParameters",
   "XCenter" : "0mm",
   "YCenter" : "0mm",
   "ZCenter" : "0mm",
   "WhichAxis" : "Z",
   "Height" : "2mm",
   "BottomRadius" : "1.56204993518133mm",
   "TopRadius" : "0mm"
   },
   {
   "NAME" : "Attributes",
   "Name" : "Cone1",
   "Flags" : "",
   "Color" : "(132 132 193)",
   "Transparency" : 0,
   "PartCoordinateSystem": "Global",
   "UDMId" : "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "MaterialValue" : "\"vacuum\"",
    "SolveInside" : True
    }
    )
```

Sample Script: IronPython script to create a cone

Create geometry and then create a grid from it using copy/paste/move

The following script demonstrates slightly more advanced use of scripting and the use of return values from script methods. It creates a 5x5 grid of cones and also demonstrates the adding of information messages to the application's message window.

```
oProject = oDesktop.GetActiveProject()
oDesign = oProject.InsertDesign("HFSS","Hersheys Kis-
ses","DrivenModal","")
oEditor = oDesign.SetActiveEditor("3D Modeler")


# create the first cone
AddInfoMessage("Creating first cone")
firstConeName = "firstCone"
coneBotRad = "1.5mm"
oEditor.CreateCone(
    {
    "NAME" : "ConeParameters",
    "XCenter" : "0mm",
    "YCenter" : "0mm",
    "ZCenter" : "0mm",
    "WhichAxis" : "Z",
    "Height" : "2mm",
    "BottomRadius": coneBotRad,
    "TopRadius" : "0mm"
    },
    {
    "NAME" : "Attributes",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Name" : firstConeName,
    "Flags" : "",
    "Color" : "(132 132 193)",
    "Transparency" : 0,
    "PartCoordinateSystem": "Global",
    "UDMId" : "",
    "MaterialValue" : "\"vacuum\"",
    "SolveInside" : True
    }
    )

# Now replicate this a few times and create an array out of it
AddInfoMessage("Replicating it 24 times")
for x in range(5):
    for y in range(5):
        # leave the first one alone in it's created
        # position
        if x == 0 and y == 0:
          continue

 # all other grid positions, replicate from the
 # first one

 # copy first
 oEditor.Copy(
    {
  "NAME" : "Selections",
  "Selections" : firstConeName
    }
 )
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# paste it and capture the pasted name
# the pasted names come in an array as we could
# be pasting a selection cmposed of multiple objects
pasteName = oEditor.Paste()[0]

# now move the pasted item to it's final position
oEditor.Move(
   {
  "NAME" : "Selections",
  "Selections" : pasteName
   },
   {
  "NAME" : "TransalateParameters",
  "CoordinateSystemID" : -1,
  "TranslateVectorX" : "%d * 3 * %s" % (x, coneBotRad),
  "TranslateVectorY" : "%d * 3 * %s" % (y, coneBotRad),
  "TranslateVectorZ" : "0mm"
   }
)

# Now fit the display to the created grid
oEditor.FitAll()
```

Related Topics

Introduction to IronPython

Scripting Using Iron Python: Putting it all Together

Creating User Defined Primitives and User Defined Models in Python Scripts

You can create User Defined Primitives and User Defined Models in Python scripts (based on the IronPython implementation).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Advantages Compared to C++

- No need to create and build project; all you need to do is create a Python script
- Python script is platform independent
- Scripts can inherit functionality from existing scripts
- Garbage collector - no need to free memory
- Easy debugging

Changes compared to C

Though methods, constants and structures are kept as close to the C implementation as possible, some changes had to be made to make code Python-compatible.

Structures

- Structures have the same names as in C implementation.
- Structures fields names are capitalized.
- Arrays in structures become lists in Python (Technically a.NET IList container)
- Structure instances are created using the supplied constructors and members are accessed using the provided access methods.

For a complete list of structures and examples please see UDP/UDM Structures.

Return Values for UDM and UDP Functions

For information on return values for each UDM and UDP function, see the Return Values section.

Constants

Enumeration/Enum constants have almost the same names as in C but the enum must be qualified by the type. Additionally, redundant "UDP", "UDM" or type prefixes have been removed. This allows for better human-readability.

```
# Example of specifying the LengthUnit enum by qualifying it

# with the type of the enum: UnitType

unitType = UnitType.LengthUnit
```

For a complete list of enum constants please see UDP/UDM Constants.

Methods

Methods are described in IUDPExtension methods, IUDMExtension methods, and UDMFunctionLibrary listed further in this document. A separate chapter includes a UDP IronPython example of fillet and chamfer.

The main differences in functions parameters (from C implementation):

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- functions names in UDPFunctionLibrary and UDMFunctionLibrary are capitalized
- arrays become a python list of objects
- `void * callback parameter` is dropped from the parameter list
- output parameters (pointer types that are filled during the function call) usually become return values
- 'list size' parameter usually will be omitted as redundant

Output Parameters

The rule for the output parameters is as follows:

- If the function has one output parameter variable and no return value, the variable will become function's return value.The same will happen if the return value is a 'success/failure' boolean ('None' will be returned on failure and parameter variable - on success).
- If the function has one output parameter and a return value, the function will return a Python tuple where function return value will be the first one in the tuple.
- If there is more than one out variable, the function will return a Python tuple with all output parameters in the specified order. If function has a return value, it must always be the first in the tuple.

```
# one output parameter; return value is ignored

udmDefinition = udmFunctionLibrary.GetDefinition()


# one output parameter; return value must be preserved. return
# and output values are packed into the return tupe, in order

(lRet, partIdsList) = udpFunctionLibrary.DetachFaces(nPartIds, faceId-
sList)

# Two output parameter; return value must be preserved

# the return tuple is (returnVal, output1, output2)

(bRet, udpPositionLow, udpPositionHigh) = udmFunc-
tionLibrary.GetBoundingBox(partId,exact);
```

Comparison with C function:

C	Python
bool getDefinition(UDMDefinition* udmDefinition, void* callbackData); where udmDefinition is an output	udmDefinition = udmFunctionLibrary.GetDefinition() (Note: callbackData is omitted in py interface)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

C	Python
parameter	
long detachIFaces(int nFacesAndPartIds, long* faceIds, long* partIds, void* callbackData); where partIds is an output parameter	(bRet, partIds) = udmFunctionLibrary.DetachIFaces (nFacesAndPartIds, faceIds) (Note: callbackData is omitted in py interface)

'List Size' Parameters

The rule for the 'list size' is as follows:

- If function has input 'List' parameter and input 'list size' parameter, 'list size' parameter will be omitted.
- If function has output 'List' parameter and output 'list size' parameter, 'list size' parameter will be omitted.
- If function has output 'List' parameter and input 'list size' parameter, 'list size' parameter won't be omitted as it's needed for memory allocation in the corresponding C++ function from the UDP/UDM function library.

Example:

```
# input list, input list size
lret = udpFunctionLibrary.Unite(objectIds)


# output list, output list size
faceIdList = udmFunctionLibrary.GetAllFaces(PartId)


# output list, input list size
(lret, partIdList) = udpFunctionLibrary.DetachFaces(listSize,
faceIdList)
```

Comparison with C function:

C	Python
bool getAllFaces(long partId,	faceIds = udmFunc-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

C	Python
long* numFaces, long** faceIds, void* callbackData); where numFaces and faceIds are output parameters and numFaces is the size of faceId.	tionLibrary.GetAllFaces(partId) (ignore numFaces as redundant: folded into faceIds, return value is omitted: folded into the faceIds is None check callbackData is omitted)
long unite(long numObjects, long* objectIds, void* callbackData); where numObjects and objectIds are input parameters and numObjects is the size of objectIds.	lret = udpFunctionLibrary.Unite (objectIds) (ignore numObjects as redundant: folded into objectIds callbackData is omitted)
long detachFaces(long nSize, long* faceIds, long* partIds, void* callbackData); where partIds is and output list and nSize is an input parameters and nSize is the size of partIds.	(lret, partIdList) = udpFunctionLibrary.DetachFaces(nSize, faceIds) (nSize is not ignored, callbackData is omitted)

Added Parameters

There is a special case in UDPFunctionLibrary: two functions - DuplicateAlongLine and DuplicateAroundAxis - have new integer listSize parameter added to their signatures.

This parameter defines the size of the output List. This is done for compliance with C++ geometry library as the size of the List must be predefined and this size is different from the existing parameter's values.

Example:

```
(ret, cloneIDs) = funcLib.DuplicateAlongLine(partID, transVec,
numCubes, cloneIdsSize)

(ret, cloneIDs) = funcLib.DuplicateAroundAxis(partID, axis, angle,
nClones, cloneIdsSize)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Here cloneIdsSize is a new integer parameter.

Comparison with C function:

C	Python
long duplicateAlongLine(long partId, UDPVector transVector, int nClones, long* nClones, void* callbackData);	(lret, cloneIds) = udmFunctionLibrary.DuplicateAlongLine(partId, transVec, nClones, cloneIdsSize) (callbackData is omitted cloneIdsSize is a new parameter)
long duplicateAroundAxis(long partId, UDPCoordinateSystemAxis axis, double angle, int nClones, long* nClones, void* callbackData);	(lret, cloneIds) = udmFunctionLibrary.DuplicateAroundAxis (partId, axis, angle, nClones, cloneIdsSize) (callbackData is omitted cloneIdsSize is a new parameter)

Developing a UDM/UDP

Creation

To create a User Defined Primitive in Python you write a Python script that implements UDPExtension class. To create a User Defined Model in Python you write a Python script that implements UDMExtension class (see links for full description).

Location

The scripts are located the same way the C based UDM/UDP are. They are expected to be under the UserDefinedParts or UserDefinedModels sub-directories of one of the library folders (SysLib, UserLib or PersonalLib). They will then appear under the appropriate menu items: **Draw** > **User Defined Primitives for UDP** or **Draw** > **User Defined Model for UDM**.

The sub-directories structure created in one of the specified directory will be displayed in the UDP/UDM menu.

Keep in mind that there is no difference between the menu display for C and Python implementations of UDM or UDP - only the file names without extensions are displayed

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Organize

"Lib" sub-directory is a special directory. The contents of this directory is not shown in the menu. In the "Lib" directory you can create Python scripts with base classes and utilities to be used in UDP/UDM Python scripts. All the Lib directories upstream of a script (till the UserDefinedModels or UserDefinedPrimitives) are included in the Python search path and this allows for easy import of helper modules in such directories.

To use UDM data structures, constants, and/or classes in your Lib sub-directory scripts you have to add import statement to the scripts:

For UDM:extension:

```
from UDM import *
```

For UDP:extension:

```
from UDP import *
```

Edit/Reload

Python is a scripting language, so if you have errors in your script, you will see them at the time you try to run the script. The errors will be displayed in the Message Manager Window. If you need more information, you might be able to get it from log files. See: Debug Logging.

You can always change your script, call **Update Menu** command from **Draw** > **User Defined Model** > **menu** or **Draw** > **User Defined Primitives** > **menu** and run the script again. If you delete script you might want to restart the application instead of calling **Update Menu**.

UDPExtension

Import

You do not have to add import statements for the predefined classes, structures, and constants - it is done for you and all data types described in this document can be used in your Python script.

However you have to add import statements to your helper scripts in your Lib sub-directory.

```
from UDP import *
```

Main class: UDPExtension

You must write a class derived from IUDPExtension with a mandatory name UDPExtension:

```
class UDPExtension(IUDPExtension):
```

The class should implement IUDPextension methods described in the topic that follows.

IUDPExtension Methods

All methods are same as the methods in the C UDP implementation. The changes to the methods signatures are just to conform to the Python style.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Mandatory Methods

These methods must be implemented in the UDP Python script as methods of UDPExtension class.

GetLengthParameterUnits()

- returns string.

GetPrimitiveTypeInfo()

- returns UDPPrimitiveTypeInfo.

GetPrimitiveParametersDefinition2()

- returns a list of UDPPrimitiveParameterDefinition2 or None on failure

AreParameterValuesValid2(errorMsg, udpParams)

- errorMsg is a c# list of strings
- udpParams is a c# list of UDPParam
- returns True if udpParams are valid, False otherwise.

CreatePrimitive2(funcLib, udpParams)

- funcLib is UDMFunction library
- udpParams is a c# list of UDPParam
- returns True on success, False on failure.

Optional Methods

These methods, which have default implementations, can be implemented as methods of UDPExtension class as needed. Default methods will return NULL or FALSE depending on the return type.

GetPrimitiveParameters()

- returns Python list of strings or NULL

GetRegisteredFaceNames()

- returns Python list of strings or NULL

GetRegisteredEdgeNames()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- returns Python list of strings or NULL

GetRegisteredVertexNames()

- returns Python list of strings or NULL

ProjectParametersOnToValidPlane2(currentUDPParams, projectedUDPParams)

- currentUDPParams is a list of UDPParam
- projectedUDPParams is a list of UDPParam
- returns True on success, False on failure.

MapParametersDefinitionVersions2(oldVersion, oldUDPParams)

- oldVersion is a string
- oldUDPParamsis a list of UDPParam
- returns Python list of UDPParam or NULL

GetOldPrimitiveParametersDefinition2(version)

- version is a string
- returns a list of UDPPrimitiveParameterDefinition2 or None on failure.

Example UDP

```
import sys
class UDPExtension(IUDPExtension):

  def GetLengthParameterUnits(self):
    return "mm"

  def GetPrimitiveTypeInfo(self)
    typeInfo = UDPPrimitiveTypeInfo(
      name = "SampleUDP",
      purpose = "example",
      company="Ansys",
      date="12.21.12",
      version = "1.0")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    return typeInfo
...
...
```

UDMExtension

Import

You do not have to add import statements for the predefined classes and structures - it is done for you, and all data types described in this document can be used in your Python script.

However you have to add import statements to your helper scripts in your Lib sun-directory.

```
from UDM import *
```

Main class: UDMExtension

You must write a class derived from IUDMExtension with a mandatory name UDMExtension:

```
class UDMExtension(IUDMExtension):
```

The class should implement IUDMExtension methods described below.

IUDMExtension Methods

All methods are the same as the methods in the C UDM implementation. The changes to the methods signatures are just to conform to the Python style.

Mandatory Methods

These methods must be implemented in the UDM Python script as methods of UDMExtension class.

GetInfo()

- returns UDMInfo object populated with appropriate UDM information.

IsAttachedToExternalEditor()

- returns True if UDM dll is attached to external editor.
- In case of python UDMs, this should typically return False

CreateInstance(funcLib)

- funcLib is UDMFunctionLibrary
- returns UDMParameters.

GetUnits(instanceId)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- instanceId is an integer.
- returns string containing units for the instance.

Refresh(funcLib, udmInParams, updatedParams, refreshModifiedPartsOnly, nonEditedPartRefIds)

This method is called every time a UDM is refreshed. Geometry creation/refresh should happen in this method.

- funcLib is UDMFunctionLibrary
- udmInParams is a list of UDMParameters that comes from desktop
- updatedParams: UDM script can change the UDM parameters it receives. Updated parameters need to be sent back to desktop. If the UDM script is not going to change any of the parameters that it received, it needs to copy udmInParams to updatedParams.
- refreshModifiedPartsOnly is a Boolean

  Supporting this flag is optional. For UDMs where the refresh performance is not an issue, it is recommended to ignore this flag and update all parts every time.

  This flag can be used to optimize performance of Refresh method when the model created by UDM is large. If the UDM consists of multiple parts, and new parameters change only a few parts amongst them, UDM script can only modify parts that are changed by the new parameters.
- nonEditedPartRefIds: If RefreshModifiedPartsOnly is true and the UDM script supports partial update, Refresh method needs to return ids of parts that are unchanged.

  returns True on success, False on failure.

ReleaseInstance(instanceId)

- instanceId is an integer.
- This should release any resources assigned to this particular instance of UDM.
- returns True on success, False on failure.

GetAttribNameForEntityId()

- Returns string that acts as a the name of the attribute containing entity IDs.
- For example, it can return a unique string such as "ATTRIB_XACIS_ID"
- Python UDMs should implement this method.

GetAttribNameForPartId()

- Returns a string that acts as a the name of the attribute containing entity IDs.
- For example, it can return a unique string such as "ATTRIB_XACIS_ID" (Can be same as

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAttribNameForEntityId())

- Python UDMs should implement this method.

Optional Methods

These methods have default implementations (default is to return NULL or FALSE depending on the return type) but can be overridden by the user as needed as methods of UDMExtension class.

DialogForDefinitionOptionsAndParams(self, defData, optData, params):

Replaces the old UDMDialogForDefinitionAndOptions method, which is still supported, but users are urged to use UDMDialogForDefinitionOptionsAndParams. If both methods are present, application will use UDMDialogForDefinitionOptionsAndParams.

- UDM can open a dialog box for UDM definition, options, parameters in this method. Definition, options, and parameters are set/modified by user and returned to application. Dll can also just give default definition, options and parameters.
- Returns two Booleans and a string
  - First Boolean returns whether the method was successful or not.
  - Second Boolean returns whether the application should open a dialog box. If it is True, application will populate a dialog box with definition, options, parameters that are returned.
  - String returned contains length units for parameters.

DialogForDefinitionAndOptions(self, defData, optData) [Deprecated]

UDM can open a dialog box for UDM definition and options in this method. Definition, and options are set/modified by user and returned to application. Dll can also just give default definition and options.

- Returns two Booleans.
  - First Boolean provides whether the call to this method was successful or not.
  - Second Boolean determines whether the application should pop up a dialog box. If this is true, application will populate the dialog box with the definitions and options that are returned. As no parameters are returned, no parameters are shown in this dialog box.

GetInstanceSourceInfo(instanceId)

- instanceId is an integer.
- returns string containing source information of UDM instance. It is used to create initial name for UDM instance.

ShouldAttachDefinitionFilesToProject()

- returns True if any of definition files needs to be attached to project
- returns a Python list of strings containing definition names of files or NULL

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example UDM

```
class UDMExtension(IUDMExtension):


  def IsAttachedToExternalEditor(self):
    return False


  def GetInfo(self)
    udmInfo = UDMInfo(
      name = "SampleUDM",
      purpose = "udm example",
      company="Ansys",
      date="12.21.12",
      version = "1.0")


    return udmInfo
  ...
  ...
```

UDMFunctionLibrary

UDMFunctionLibrary implements IUDMFunctionLib interface. The IUDMFunctionLib object is passed as a parameter to Python script in the following functions

- CreateInstance
- Refresh

You can call any of the functions from the functions list (shown below).

```
partRefId = udmFunctionLib.GetPartRefId(partId)
```

For example sample code that calls GetBoundingBox in Python script can look like this:

```
partId = 10

exact = True

udpPosition = UDPPosition(0,0,0)


(bret, udpPositionLow, udpPositionHigh) = udmFunc-
tionLibrary.GetBoundingBox(partId, exact);
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
if bret:
   udpPosition.X = udpPositionLow.X
```

As you can see udpPositionLow and udpPositionHigh output parameters are defined in the call to GetBoundingBox function. There is no need to define them before the function call.

Functions list:

1. ***List_of_UDMDefinition:*** udmDefinitionList = **GetDefinition**()
2. ***List_of_UDMOption:*** udmOptionList = **GetOptions**()
3. ***bool:*** bret = **SetMaterialName**(***string:*** matName, ***int:*** partId)
4. ***bool:*** bret = **SetMaterialName2**(***string:*** matName, ***string:*** partName)
5. ***bool:*** bret = **SetPartName**(***string:*** partName, ***int:*** partId)
6. ***int:*** iret = **GetInstanceId**()
7. ***string:*** str = **GetPartRefId**(***int:*** partId)
8. ***bool:*** bret = **SetPartRefId**(***int:*** partId, ***string:*** refId)
9. ***List_of_int:*** faceIds = **GetAllFaces**(***int:*** partId)
10. ***List_of_int:*** edgeIds = **GetAllEdges**(***int:*** partId)
11. ***List_of_int:*** vertexIds = **GetAllVertices**(***int:*** partId)
12. ***bool:*** bret = **SetFaceAttribs**(***List_of_int:*** faceIds, ***List_of_string:*** attribs)
13. ***bool:*** bret = **SetEdgeAttribs**(***List_of_int:*** edgeIds, ***List_of_string:*** attribs)
14. ***bool:*** bret = **SetVertexAttribs**(***List_of_int:*** vertexIds, ***List_of_string:*** attribs)
15. ***string:*** str = **GetModelerUnit**()
16. ***string:*** str = **GetCacheFileForUDMResume**()
17. ***bool:*** bret = **SetPartColor**(***int:*** partId, ***int:*** nColor)
18. ***bool:*** bret = **SetPartFlags**(***int:*** partId, ***int:*** nFlags)
19. (***bool:*** bret, ***UDPPosition:*** low, ***UDPPosition:*** high) = **GetBoundingBox**(***int:*** partId, ***bool:*** exact)
20. ***bool:*** bret = **IsParametricUpdate**()
21. ***bool:*** bret = **SetMaterialNameByRefId**(***string:*** partRefID, ***string:*** matName)
22. ***bool:*** bret = **SetPartNameByRefId**(***string:*** partRefId, ***string:*** partName)
23. ***bool:*** bret = **SetPartColorByRefId**(***string:*** partRefId, ***int:*** nColor)
24. ***bool:*** bret = **SetPartFlagsByRefId**(***string:*** partRefId, ***int:*** nFlags)

In addition to the above functions all functions defined in the UDPFunctionLib are available in the IUDMFunctionLib and can be called directly exactly the same way as the IUDMFunctionLib functions.

Example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
udmFunctionLib.CreateCircle(center,radius,ratio,isCovered)
```

UDM/UDP Functions

Return Values for Each UDM and UDP Function

ID – ID of created Object

SI – Success Indicator. Identifies whether or not operation was successful.

Functions list:

1. ***bool***: SI = **AddMessage(*MessageSeverity*:** messageSeverity, ***string***: message**)**
2. ***bool***: SI = **NameAFace(*UDPPosition***: pointOnFace, ***string***: faceName**)**
3. ***bool***: SI = **NameAEdge(*UDPPosition***: pointOnEdge, ***string***: edgeName**)**
4. ***bool***: SI = **NameAVertex(UDPPosition**: pointOnVertex, ***string***: vertexName**)**
5. ***int***: ID = **GetFaceIDFromPosition(*UDPPosition***: pointOnFace**)**
6. ***int***: ID = **GetEdgeIDFromPosition(*UDPPosition***: pointOnEdge**)**
7. ***int***: ID = **CreatePolyline(*UDPPolylineDefinition***: polylineDefinition**)**
8. ***int***: ID = **CreateRectangle(*CoordinateSystemPlane***: whichPlane, ***UDPPosition***: centerPoint, ***List_of_double***: widthAndHeight, ***int***: isCovered**)**
9. ***int***: ID = **CreateArc(*CoordinateSystemPlane***: whichPlane, ***UDPPosition***: centerPoint, ***UDPPosition***: startPoint, ***double***: fAngle**)**
10. ***int***: ID = **CreateCircle(*CoordinateSystemPlane***: whichPlane, ***UDPPosition***: centerPoint, ***double***: fRadius, ***int***: isCovered**)**
11. ***int***: ID = **CreateEllipse(*CoordinateSystemPlane***: whichPlane, ***UDPPosition***: centerPoint, ***double***: fMajorRadius, ***double***: fRadiusRatio, ***int***: isCovered**)**
12. ***int***: ID = **CreateRegularPolygon(*CoordinateSystemPlane***: whichPlane, ***UDPPosition***: centerPoint, ***UDPPosition***: startPoint, ***int***: numOfSides, ***int***: isCovered**)**
13. ***int***: ID = **CreateEquationBasedCurve(*UDPEquationBasedCurveDefinition***: curveDefinition**)**
14. ***int***: ID = **CreateEquationBasedSurface(*UDPEquationBasedSurfaceDefinition***: surfaceDefinition**)**
15. ***int***: ID = **CreateSpiral(*UDPSpiralDefinition***: spiralDefinition**)**
16. ***int***: ID = **CreateBox(*UDPPosition***: startPoint, ***List_of_double***: boxXYZsize**)**
17. ***int***: ID = **CreateSphere(*UDPPosition***: centerPoint, ***double***: fRadius**)**
18. ***int***: ID = **CreateCylinder(*CoordinateSystemAxis***: whichAxis, ***UDPPosition***: centerPoint, ***double***: fRadius, ***double***: fHeight**)**
19. ***int***: ID = **CreateCone(*CoordinateSystemAxis***: whichAxis, ***UDPPosition***: centerPoint, ***double***: fBottomRadius, ***double***: fTopRadius, ***double***: fHeight**)**
20. ***int***: ID = **CreateTorus(*CoordinateSystemAxis***: whichAxis, ***UDPPosition***: centerPoint, ***double***: fMajorRadius, ***double***: fMinorRadius**)**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

21. ***int***: ID = **CreatePolyhedron(*CoordinateSystemAxis***: whichAxis, ***UDPPosition***: centerPoint, ***UDPPosition***: startPosition, ***int***: numOfSides, ***double***: fHeight**)**
22. ***int***: ID = **CreateHelix(*UDPHelixDefinition***: helixDefinition**)**
23. ***bool***: SI = **Unite(*List_of_int***: pObjectIDArray**)**
24. ***bool***: SI = **Subtract(*List_of_int***: pBlankObjectIDArray, ***List_of_int***: pToolObjectIDArray**)**
25. ***bool***: SI = **Intersect(*List_of_int***: pObjectIDArray**)**
26. ***bool***: SI = **Imprint(*List_of_int***: pBlankObjectIDArray, ***List_of_int***: pToolObjectIDArray**)**
27. ***bool***: SI = **SweepAlongVector(*int***: profileID, ***UDPVector***: sweepVector, UDPSweepOptions: sweepOptions**)**
28. ***bool***: SI = **SweepAroundAxis(*int***: profileID, ***CoordinateSystemAxis***: whichAxis, ***double***: sweepAngle, ***UDPSweepOptions***: sweepOptions**)**
29. ***bool***: SI = **SweepAlongPath(*int***: profileID, ***int***: pathID, ***UDPSweepOptions***: sweepOptions**)**
30. ***bool***: SI = **Translate(*int***: partID, ***UDPVector***: translateVector**)**
31. ***bool***: SI = **Rotate(*int***: partID, ***CoordinateSystemAxis***: whichAxis, ***double***: rotateAngle**)**
32. ***bool***: SI = **Mirror(*int***: partID, ***UDPPosition***: mirrorPlaneBasePosition, ***UDPVector***: mirrorPlaneNormalVector**)**
33. ***bool***: SI = **Transform(*int***: partID, ***List_of_double***: rotationMatrix, ***UDPVector***: translateVector**)**
34. ***bool***: SI = **Scale(*int***: partID, ***double***: xScale, ***double***v: yScale, ***double***: zScale**)**
35. (***bool***: SI, ***List_of_int***: cloneIDs**)** = **DuplicateAlongLine(*int***: partID, ***UDPVector***: translateVector, ***int***: numTotalObjs, ***int***: cloneIDsListSize**)**
36. (***bool***: SI, ***List_of_int***: cloneIDs) = **DuplicateAroundAxis(*int***: partID, ***CoordinateSystemAxis***: whichAxis, ***double***: rotateAngle, ***int***: numTotalObjs, ***int***: cloneIDsListSize**)**
37. ***int***: ID = **DuplicateAndMirror(*int***: partID, ***UDPPosition***: mirrorPlaneBasePosition, ***UDPVector***: mirrorPlaneNormalVector**)**
38. ***bool***: SI = **Connect(*List_of_int***: objectIDArray**)**
39. ***bool***: SI = **Offset(*int***: partID, ***double***: offsetDistance**)**
40. ***int***: ID? = **Section(*int***: partID, ***CoordinateSystemPlane***: sectionPlane**)**
41. (***bool***: SI , ***int***: ID) = **Split(*int***: partID, ***CoordinateSystemPlane***: splitPlane, **SplitWhichSideToKeep**: whichSideToKeep, ***bool***: bSplitCrossingObjectsOnly**)**
42. (***bool***: SI , ***List_of_int***: importedObjectIDs) = **ImportNativeBody2(*string***: fileNameWithFullPath**)**
43. (***bool***: SI, ***List_of_int***: importedObjectIDs) = **ImportAnsoftGeometry(*string***: fileNameWithFullPath, ***List_of_string***: overridingParamsNameArray, ***List_of_UDPParam***: overridingParamsArray**)**
44. ***int***:ID = **Clone(*int***: partID**)**
45. ***bool***: SI = **DeletePart(*int***: partID**)**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

46. ***int***: ID = **CreateObjectFromFace(*int*: faceID)**
47. ***bool***: SI = **Fillet(*UDPBLNDElements*: entitiesToFillet, *UDPBLNDFilletOptions*: filletOptions)**
48. ***bool***: SI = **Chamfer(*UDPBLNDElements*: entitiesToChamfer, *UDPBLNDChamferOptions*: chamferOptions)**
49. (***bool***: SI, ***List_of_int***: newPartIDs) = **DetachFaces(*int*: newPartIDArraySize, *List_of_int*: faceIDs)**
50. (***bool***: SI, ***List_of_int***: newPartIDs) = **DetachEdges(*int*: newPartIDArraySize, *List_of_int*: edgeIDs)**
51. ***int***: ID = **CreateObjectFromEdge(*int*: edgeID)**
52. ***bool***: SI = **SheetThicken(*int*: partID, *double*: fThickness, *bool*: bThickenBothSides)**
53. (***bool***: SI , ***List_of_int***: newPartIDArray) = **SweepFaceAlongNormal(*int*: newPartIDArraySize, *List_of_int*: faceIDArray, *double*: sweepLength)**
54. ***bool***: SI = **CoverLine(*int*: partID)**
55. ***bool***: SI = **CoverSurface(*int*: partID)**
56. ***bool***:SI= **UncoverFaces(*List_of_int*: faceIDArray)**
57. (***bool***: SI , ***int***: numPartsCreated, ***List_of_int***: faceIDArray) = **SeparateBodies(*int*: partID, *int*: numPartsCreated)**
58. ***bool***: SI = **MoveFaces(*List_of_int*: faceIDArray, *bool*: bMoveAlongNormal, *double*: fOffsetDistance, *UDPVector*: moveVector)**
59. ***bool***: SI = **WrapSheet(*int*: sheetBodyID, *int*: targetBodyID)**
60. ***bool***: SI = **ImprintProjection(*int*: blankBodyID, *List_of_int*: toolBodyIDArray, *bool*: bNormalProjection, *UDPVector*: projectDirection, *double*: projectDistance)**
61. ***string***: path = **GetTempDirPath()**
62. ***string***: path = **GetSysLibDirPath()**
63. ***string***: path = **GetUserLibDirPath()**
64. ***string***: path = **GetPersonalLibDirPath()**
65. ***string***: path = **GetInstallDirPath()**
66. ***string***: path = **GetProjectPath()**
67. (***bool***: SI, ***bool***: abort) = **SetProgress(*UDPProgress*: progress)**

UDP/UDM Structures and Constants

The following sections describe:

- UDP/UDM Structures
- UDP/UDM Constants

UDP/UDM Structures

Differences compared to C API:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- **UDMDefinition**
- **UDMOptions**
- **UDMParameters**

  Instead of containing arrays of data, the structures contain single fields where each field corresponds to an item in a different array from the original C API. The structure objects thus constructed are added to the Python list. Alternately the Python list can be initialized using the structure objects.

  Example (creating UDMParameter list):

```
udmParamList = [
  UDMParameter(
    "cubeSizeName", UnitType.LengthUnit,
    UDPParam(ParamDataType.Double, cubeSize),
    ParamPropType.Value,
    ParamPropFlag.MustBeReal),
  UDMParameter(
    "cubeDistanceName", UnitType.LengthUnit,
    UDPParam(ParamDataType.Double, cubeDistance),
    ParamPropType.Value,
    ParamPropFlag.MustBeReal),
  UDMParameter("numCubesName", UnitType.LengthUnit,
    UDPParam(ParamDataType.Int, numCubes),
  ParamPropType.Number,
  ParamPropFlag.MustBeInt]
```

- **UDPParam**
- **UDPParamData**

  Data field in UDPParam is now an object - the same for all types of data used - as Python can work with any type of data.

  UDPParamData is obsolete, thus not implemented. Be sure to set proper data type to UDPParam.DataType when setting UDPParam.Data.

  Example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
nCubesParam = UDPParam(ParamDataType.Int, numCubes)
nCubes = nCubesParam.Data


distanceParam = UDPParam()
distanceParam.setDouble(10.5)
doubleDistance = distanceParam.Data * 2
```

- **UDP3x3Matrix**

  The structure is not implemented. Use size 9 Python List of doubles instead.

  Example:

```
rotationMatrix =[0,0,1, 1,0,0, 0,0,1]
udpFunctionLib.Transform(partId, rotationMatrix, trans-
lationVector)
```

List of Structures

You can use constructors to create a structure. You can also modify fields - directly or by provided methods.

Example:

```
pos1 = UDPPosition(1,2,3)
pos2 = UDPPosition(x=1,y=10,z=0)
pos2.Z = pos1.Z
udpParam = UDPParam(ParamDataType.Double,1)
value = udpParam.Data
```

Structure	Construction	Members
UDPPrimitiveTypeInfo	UDPPrimitiveTypeInfo(string name, string purpose, string company, string date, string version)	string Name string Purpose string Company string Date string Version
UDPPrim-itiveParameterDefinition	UDPPrim-itiveParameterDefinition(	string Name string Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	string name, string description, UnitType unitType, double defaultValue)	UnitType UnitType double DefaultValue
UDPParam	UDPParam() UDPParam(ParamDataType dataType, object data) object can be int, double, string, bool or UDPPosition **methods:** setInt(int val) setBool(bool val) setString(string val) setDouble(double val) setPosition(UDPPosition val)	ParamDataType DataType object Data object can be int, double , string, bool or UDPPosition
UDPPrim-itiveParameterDefinition2	UDPPrim-itiveParameterDefinition2(string name, string description, UnitType unitType, ParamPropType propType, ParamPropFlag propFlag, UDPParam defaultValue)	string Name string Description UnitType UnitType ParamPropType PropType ParamPropFlag PropFlag UDPParam DefaultValue
UDPPosition	UDPPosition(double x, double y,	double X double Y double Z

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	double z)	
UDPVector	UDPVector(double x, double y, double z)	double X double Y double Z
UDPSweepOptions	UDPSweepOptions(SweepDraftType draftType, double draftAngle, double twistAngle)	SweepDraftType DraftType double DraftAngle double TwistAngle
UDPPolylineSegmentDefinition	UDPPolylineSegmentDefinition(PolylineSegmentType segmentType, int segmentStartIndex, int numberOfPoints, double angle, UDPPosition centerPoint, CoordinateSystemPlane arcPlane)	PolylineSegmentType SegmentType int segmentStartIndex, int numberOfPoints, double angle, UDPPosition centerPoint, CoordinateSystemPlane arcPlane)
UDPPolylineDefinition	UDPPolylineDefinition() UDPPolylineDefinition(List_of_UDPPosition positions, List_of_UDPPolylineSegmentDefinition segDefs, int closed, int covered)	int IsClosed int IsCovered List_of_UDPPosition ArrayOfPosition List_of_UDPPolylineSegmentDefinition ArrayOfSegmentDefinition
UDPEquationBasedCurveDefinition	UDPEquationBasedCurveDefinition(string functionXt, string functionYt, string functionZt,	string FunctionXt string FunctionYt string FunctionZt double TStart double TEnd

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	double tStart, double tEnd, int numOfPointsOnCurve)	int NumOfPointsOnCurve
UDPEquationBasedSurfaceDefinition	UDPEquationBasedSurfaceDefinition(string functionXuv, string functionYuv, string functionZuv, double uStart, double uEnd, double vStart, double vEnd int reserved1 int reserved2)	string FunctionXuv string FunctionYuv string FunctionZuv double UStart double UEnd double VStart double VEnd two integer arguments that are reserved for future use. They need to be provided, for example as 0. For example: `theSurfaceDefinition = UDPEquationBasedSurfaceDefinition ("u","v","1",0,1,0,1,0,0)`
UDPHelixDefinition	UDPHelixDefinition(int profileID, UDPPosition ptOnAxis, UDPPosition axisDir, double noOfTurns, bool isRightHanded, double radiusChangePerTurn, double pitch)	int ProfileID UDPPosition PtOnAxis UDPPosition AxisDir double NoOfTurns bool IsRightHanded double RadiusChangePerTurn double Pitch
UDPSpiralDefinition	UDPSpiralDefinition(int profileID, UDPPosition ptOnAxis, UDPPosition axisDir, double noOfTurns,	int ProfileID UDPPosition PtOnAxis UDPPosition AxisDir double NoOfTurns bool IsRightHanded

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	bool isRightHanded, double radi-usChangePerTurn)	double RadiusChangePerTurn
UDPBLNDElements	UDPBLNDElements(int partID, int noOfEdges; int* listOfEdges;) UDPBLNDElements(int partID, int noOfVertices; int* listOfVertices;)	UDPBLNDElements can hold either edges or vertices, but not both at the same time. Edges should be applied to solids, and vertices should be applied to sheets. int PartID /* part to be blended i.e. filleted/chamfered */ int noOfEdges; int* listOfEdges; /* edges to be blended */ int noOfVertices; int* listOfVertices; /* vertices to be blended */
UDPBLNDFilletOptions	UDPBLNDFilletOptions(bool supressFillet, BLNDFilletRadiusLaw fil-letRadiusLaw, double filletStartRadius, double filletEndRadius, bool fol-lowSmoothEdgeSequence, BLNDFilletType filletType, double setbackDistance, double bulgeFactor)	bool SupressFillet /* Reserved for future */ BLNDFilletRadiusLaw Fil-letRadiusLaw double FilletStartRadius double FilletEndRadius bool Fol-lowSmoothEdgeSequence /* Reserved for future */ BLNDFilletType FilletType double SetbackDistance double BulgeFactor /* Reserved for future */
UDPBLNDChamferOptions	UDPBLNDChamferOptions(bool supressChamfer, BLNDChamferRangeLaw chamferRangeLaw,	bool SupressChamfer BLNDChamferRangeLaw Cham-ferRangeLaw double ChamferLeftRange

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	double chamferLeftRange, double chamferRightRange)	double ChamferRightRange
UDPProgress	UDPProgress(int prog, int subProg, string mesg, string subMesg)	int Prog int SubProg string Mesg string SubMesg
UDMInfo	UDMInfo(string name, string purpose, string company, string date, string version)	string Name string Purpose string Company string Date string Version
UDMDefinition	UDMDefinition() UDMDefinition(string name, UDParam value, ParamPropType propType, ParamPropFlag propFlag)	string DefName UDPParam DefValue ParamPropType PropType ParamPropFlag PropFlag
UDMOption	UDMOption() UDMOption(string name, UDParam value, ParamPropType propType, ParamPropFlag propFlag)	string OptName UDPParam OptValue ParamPropType PropType ParamPropFlag PropFlag
UDMParameter	UDMParameter() UDMParameter(	string ParamName UDPParam ParamValue UnitType UnitType

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Structure	Construction	Members
	string name, UDParam value, UnitType unitType, ParamPropType propType, ParamPropFlag propFlag)	ParamPropType PropType ParamPropFlag PropFlag

UDP/UDM Constants

Full names of enum constants must be used in scripts.

Example:

```
unitType = UnitType.LengthUnit
dataType = ParamDataType.Int
```

Enum constants:

enum Constant	Parameters
UnitType	NoUnit LengthUnit AngleUnit
ParamDataType	Int Double String Bool Position Unknown
ParamPropType	Text Menu Number Value FileName Checkbox Position Unknown
ParamPropFlag	NoFlag

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

enum Constant	Parameters
	ReadOnly MustBeInt MustBeReal Hidden Unknown
CoordinateSystemAxis	XAxis YAxis ZAxis
CoordinateSystemPlane	XYPlane YZPlane ZXPlane
SweepDraftType	ExtendedDraft RoundDraft NaturalDraft MixedDraft
SplitWhichSideToKeep	SplitKeepBoth SplitKeepPositiveOnly SplitKeepNegativeOnly
PolylineSegmentType	LineSegment ArcSegment SplineSegment AngularArcSegment
MessageSeverity	WarningMessage ErrorMessage InfoMessage IncompleteMessage FatalMessage
BLNDFilletRadiusLaw	BLNDConstantRadius BLNDVariableRadius
BLNDFilletType	BLNDRound /* The outward surface of the fillet is curved.*/

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

enum Constant	Parameters
	BLNDMitered /* The outward surface of the fillet is flat and cut at an angle.*/
BLNDChamferRangeLaw	BLNDConstantRange BLNDVariableRange
PartPropertyFlags	PropNonModel PropDisplayWireFrame PropReadOnly PostprocessingGeometry PropInvisible PropShowDirection PropDummy

UDP Python Example

This Python script example demonstrates how to use the UDPBLNDElements structure and the UDP chamfer and fillet functions.

```
import sys
primitive_info = UDPPrimitiveTypeInfo(
  name="Fillet_Chamfer",
  purpose="Fillet Chamfer Example",
  company="Ansys",
  date="09/11/2020",
  version="1.0")
primitive_param_definitions = [
  UDPPrimitiveParameterDefinition2(
    "x_size",
    "",
    UnitType.LengthUnit,
    ParamPropType.Value,
    ParamPropFlag.MustBeReal,
    UDPParam(ParamDataType.Double, 10)),
  UDPPrimitiveParameterDefinition2(
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "y_size",
      "",
      UnitType.LengthUnit,
      ParamPropType.Value,
      ParamPropFlag.MustBeReal,
      UDPParam(ParamDataType.Double, 5)),
    UDPPrimitiveParameterDefinition2(
      "z_size",
      "",
      UnitType.LengthUnit,
      ParamPropType.Value,
      ParamPropFlag.MustBeReal,
      UDPParam(ParamDataType.Double, 2))
]
length_units = "mm"


############################################################
# Class Implementation
############################################################
class UDPExtension(IUDPExtension):
  def CreatePrimitive2(self, func_lib, param_values):
    """
    Inbuilt function that is called to generate a UDP after suc-
    cessful validation
    :param func_lib: drawing inbuilt class, see in Help: UDMFunc-
    tionLibrary
    :param param_values: list of udp parameter values (user input)
    generated by UDP Core
    :return: None
    """
    param_dict = self.get_param_dict(param_values)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
start_point = UDPPosition(0, 0, 0)
box = func_lib.CreateBox(start_point, [
  param_dict["x_size"],
  param_dict["y_size"],
  param_dict["z_size"]
])

# points on the middle of 4 vertical edges
points = [
  [0, 0, param_dict["z_size"]/2],
  [param_dict["x_size"], 0, param_dict["z_size"]/2],
  [param_dict["x_size"], param_dict["y_size"], param_dict["z_
  size"]/2],s
  [0, param_dict["y_size"], param_dict["z_size"]/2]
]

edges = [func_lib.GetEdgeIDFromPosition(UDPPosition(point[0],
point[1], point[2])) for point in points]

fillet_rad = 0.1 * param_dict["x_size"] # 10% of X size
fillet_opt = UDPBLNDFilletOptions(True, BLNDFil-
letRadiusLaw.BLNDConstantRadius, fillet_rad, 0.0, True, BLNDFil-
letType.BLNDRound, 0.0, 0.0)

chamfer_length = 0.1 * param_dict["x_size"] # 10% of X size
chamfer_opt = UDPBLNDChamferOptions(False, BLNDCham-
ferRangeLaw.BLNDConstantRange, chamfer_length, 0.0)

# select your geometry to which to apply operations
blend_element = UDPBLNDElements(box)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    # specify attribute ListOfEdges to which edges to apply fillet
    operation
    blend_element.ListOfEdges = edges[0:2]
    func_lib.Fillet(blend_element, fillet_opt)

    # redeclare attribute ListOfEdges to which edges to apply chamfer
    operation
    blend_element = UDPBLNDElements(box)
    blend_element.ListOfEdges = edges[2:4]
    func_lib.Chamfer(blend_element, chamfer_opt)

    # Provide to the user Info message indicating success
    func_lib.AddMessage(MessageSeverity.InfoMessage, "Completed!")

def GetPrimitiveTypeInfo(self):
    return primitive_info

def GetLengthParameterUnits(self):
    return length_units

def GetPrimitiveParametersDefinition2(self):
    return primitive_param_definitions

def AreParameterValuesValid2(self, error, udp_params):
    return True

# Custom Functions
def get_param_value_by_name(self, param_values, param_name):
    """
    Function to get a value of a single parameter accessing it by
    name
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    :param param_values: list of udp parameter values (user input)
    generated by UDP Core

    :param param_name: name of the parameter as specified in defin-
    ition list

    :return: Value of the parameter or None if parameter does not
    exist

    """

    param_dict = self.get_param_dict(param_values)

    value = param_dict.get(param_name, None)

    return value


  def get_param_dict(self, param_values):

    """

    Function to return a dictionary of UDP parameter name and value
    (key: value) pairs

    :param param_values: list of udp parameter values (user input)
    generated by UDP Core

    :return: dict of parameter name and values

    """

    udm_param_def = self.GetPrimitiveParametersDefinition2()

    param_dict = {}

    for i, param in enumerate(udm_param_def):

      param_value = param_values[i].Data

      if str(param.PropType) != "Menu":

        param_dict[param.Name] = param_value

      else:

        param_dict[param.Name] = param_value.replace('"', '').split
        (",")[0]

    return param_dict
```

Introduction to CPython

CPython can be used to:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- Launch Ansys Electronics Desktop (InitializeNew)
- Connect with a running instance of Ansys Electronics Desktop (Initialize)
- Execute Ansys Electronics Desktop script functions

One advantage of CPython is the large set of libraries and tools that are available. See below for instructions on modifying a script so that it can be launched with CPython interpreters.

CPython Script Engine and ansysedt process are communicated using GRPC, the ansysedt process started as GRPC server, the Script Script Engine acted as the GRPC client.

Creating an External Script

While the same as IronPython when run externally, a CPython recorded script must be modified by adding the following lines to the beginning of your script *before* `import ScriptEnv`

```
import sys

  # Imports the sys module containing system-specific functions nat-
  ive to Python.

sys.path.append(r"<InstallationPath>/PythonFiles/DesktopPlugin")

  # Adds the PythonFiles/DesktopPlugin subfolder to the list of dir-
  ectories Python searches for modules and files.
```

Those lines are followed by:

```
import ScriptEnv

  # This imports ScriptEnv.py from the installation path specified
  above.
```

Follow that with:

- Either InitializeNew() *or* Initialize(), as described below.
- Any desired Electronics Desktop scripting commands.
- Closing command, as described below.

Start ansysedt as GRPC server

But default ansysedt process will be started as GRPC server. When there is no argument provided ansysedt.exe it start listening on the first available port from 50051.

-grpcsrv Flag

ansysedt -grpcsrv <optional port number or port range>.

ansysedt –grpcsrv

If no Port Number specified, same as the default like no grpcsrv flag

ansysedt –grpcsrv portNumber

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ansysedt process will be listening on the port number, but report error as the port is used by other application.

ansysedt –grpcsrv FirstPortNumber:LastPortNumber

ansysedt process will be listening on the first available port within the port range, but report error if all ports in the range used.

ansysedt –grpcsrv FirstPortNumber:NumberOfPorts

ansysedt process will be listening on the first available port within the port range, but report error if all ports in the range used.

Note: If the last number if smaller than the first number the last number will be treated as the number of ports. (50051: 50250 has the same range as 50051:200)

Connect Functions:

1. Connect(projectPath)
    a. Connect to the opened project.
    b. Launch ansysedt.exe, open the project, then connect to it.
    c. Error if the project does not exist.
2. Connect(portNumber) Connect to running ansysedt process in the local machine. Error if no ansysedt process listening on this port.
3. Connect(machine, projectPath) connect to an open project in remote Machine. projectPath must be a network path. Error if no ansysedt process opened the project.
4. Connect(machine, portNumber) connect to a running ansysedt process on the remote machine. Error if no ansysedt process listening on this port.

Example:

```
import sys

sys.path.append(r"<InstallationPath>/PythonFiles/DesktopPlugin")

import ScriptEnv

ScriptEnv.Connect(r”c:\myProjects\Project1.aedt”)

oProject = oDesktop.GetActiveProject()

print(oProject.GetName())
```

Launching Electronics Desktop

To launch a new instance of Electronics Desktop and connect oApplication and oDesktop to it:

```
InitializeNew(NonGraphical = <True|False>, Module = None, Machine =
"", Port = <Port#>)
```

Where:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- **NonGraphical** – Specifies whether to launch Electronics Desktop in non-graphical mode.
- **Module** – Behavior remains unchanged from Iron Python and should be left defaulted to "None." See the code in ScriptEnv.py for more details.
- **Machine** – Currently an empty string, as InitializeNew() will only launch Electronics Desktop on the current machine.
- **Port** – Electronics Desktop will launch using the first unused port it finds starting at <Port#>. If Port = 0, the starting port will be 50051.

> **Note:**
>
> InitializeNew() will *always* launch a new instance of Electronics Desktop. Please use Initialize() to connect to an existing instance. See below.

Connecting with a Running Instance of Electronics Desktop

To connect oApplication and oDesktop to an existing Electronics Desktop instance, or launch a new instance and connect to it if necessary:

```
Initialize(name, NG = <True|False>, machine = "", port = <Port#>)
```

Where:

- **Name** – Ignored.
- **NG** – If launch is necessary, specifies whether to launch Electronics Desktop in non-graphical mode.
- **Machine** – The machine on which to launch/connect. For current machine, pass empty string or use localhost.
- **Port** – If port is nonzero, the script tries to connect to an existing instance on <port#> running on <machine>. If there is no instance running on that <port>, a new instance of Electronics Desktop launches on that port and then connects to it. If port = 0, the new instance is launched on the first free port, starting at 50051.

Closing Electronics Desktop/Ending the Script

To close Electronics Desktop, add the following line to the end of the script:

```
ScriptEnv.Shutdown()

   # This stops ScriptEnv.py. If you are running multiple scripts,
   include this only at the end of the last script.
```

-grpcsrv Flag

This flag will launch the application in a mode where the executable serves as a scripting server that can be used for CPython scripting in conjunction with the CPython stand alone scripting instructions that were mentioned earlier. The
-ng flag can be combined with -grpcsrv.

On Windows:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
ansysedt.exe -grpcsrv <optional port number>.
```

On Linux:

```
ansysedt -grpcsrv <optional port number>.
```

If the port number is omitted, the default of 50051 will be used.

With -grpcsrv, a message will be displayed in the **Messages** window indicating that the server was started. If the requested port is in use by another application, starting the sever may fail.

Related Topics:

Standalone Scripting in Iron Python

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

Ansys Electronics Desktop Scripting

This chapter provides an overview of scripting in Ansys Electronics Desktop.

Overview of Ansys Electronics Desktop Scripting Objects

Running a Script

Recording a Script

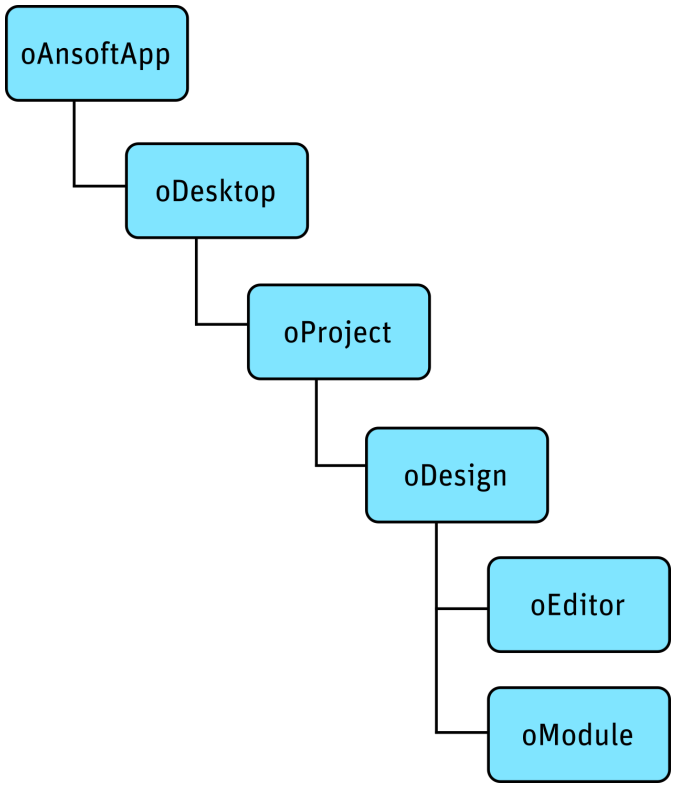
Working with Project Scripts

Executing a Script from Within a Script

Ansys Electronics Desktop Scripting Conventions

Overview of Electronics Desktop Scripting Objects

When you record a script using Ansys Electronics Desktop, the beginning of the script must contain some standard commands, as illustrated in the following chart. The commands in the chart define the objects used by Electronics Desktop in the script and assign values to these objects. The objects are used in the hierarchical order shown.



The commands are described below, followed by examples.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

oAnsoftApp

The **oAnsoftApp** object provides a handle for Iron Python to access the `Ansoft.ElectronicsDesktop` product.

In Iron Python, for example:

```
oAnsoftApp = CreateObject('Ansoft.ElectronicsDesktop')
```

oDesktop

The **oDesktop** object is used to perform desktop-level operations, including project management.

In Iron Python, for example:

```
oDesktop = oAnsoftApp.GetAppDesktop()
```

Consult the following for details about script commands recognized by `oDesktop`:

- Desktop Object Script Commands.

oProject

The **oProject** object corresponds to one project open in Electronics Desktop. It is used to manipulate the project and its data. Its data includes variables, material definitions, and one or more designs.

In Iron Python, for example:

```
oProject = oDesktop.GetActiveProject()
```

Consult the following for details about script commands recognized by `oProject`:

- Project Object Script Commands

oDesign

The **oDesign** object corresponds to a design in the project. This object is used to manipulate the design and its data, including variables, modules, and editors.

In Iron Python, for example:

```
oDesign = oProject.GetActiveDesign()
```

Consult the following for details about script commands recognized by `oDesign`:

- Design Object Script Commands
- Output Variable Script Commands
- Reporter Editor Script Commands

oEditor

The **oEditor** object corresponds to an editor, such as the 3D Modeler, Layout, or Schematic editor. This object is used to add and modify data in the editor.

In Iron Python, for example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oEditor = oDesign.SetActiveEditor('3D Modeler')
```

Consult the following for details about script commands recognized by `oEditor`:

- 3D Modeler Editor Script Commands
- Layout Editor Script Commands

> **Important:**
>
> There is no Reporter Editor object for `oEditor`. Reporter Editor commands are executed by `oDesign`.
>
> See: Reporter Editor Script Commands.

oModule

The **`oModule`** object corresponds to a module in the design. Modules are used to handle a set of related functionalities.

In IronPython, for example:

```
oModule = oDesign.GetModule('BoundarySetup')
```

Consult the following for details about script commands recognized by `oModule`:

- Analysis Module Script Commands
- Boundary and Excitation Module Script Commands
- Field Overlays Module Script Commands
- Mesh Operations Module Script Commands
- Optimetrics Module Script Commands
- Radiation Module Script Commands
- Reduce Matrix Module Script Commands
- Solutions Module Script Commands

Example Script Opening

Combining the above objects, a script in Iron Python could begin like the following:

```
oAnsoftApp = CreateObject("Ansoft.ElectronicsDesktop")
oDesktop = oAnsoftApp.GetAppDesktop()
oProject = oDesktop.SetActiveProject("Project1")
oDesign = oProject.SetActiveDesign("Design1")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oModule = oDesign.GetModule("BoundarySetup")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetActiveProject and GetActiveDesign for Wider Use

The sample script above only works for "Design1" within "Project1". To create a script that is usable for any open project, you can use one or both of `GetActiveProject` and `GetActiveDesign`.

In IronPython:

```
oProject = oDesktop.GetActiveProject()
oDesign = oProject.GetActiveDesign()
```

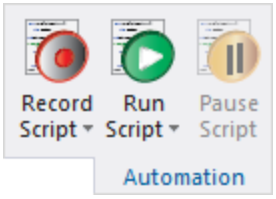
Running a Script

Electronics Desktop scripts can be run from within the software or from the command line.

Within Electronics Desktop

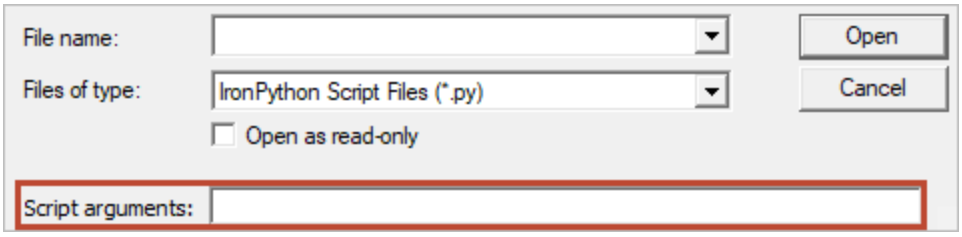
To run scripts in Electronics Desktop:

1. Click **Tools** > **Run Script**, or select the **Automation** tab and click the **Run Script** icon:



   The **Run Script** file browser appears.

2. Use the file browser to locate the script file (*.py).
3. If desired, type script arguments in the Script Arguments field:



4. Click **Open**.

   Electronics Desktop executes the script.

   While script execution is in progress, the **Run Script** button transforms into a **Stop Script** button. Click **Stop Script** to stop the script execution.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

To temporarily pause a running script, click **Pause Script**. This button transforms into a **Resume Script** button, which you can click to resume script execution.

From the Command Line

To run a script from a command line, add the `-runscriptandexit` or `-runscript` argument to the Electronics Desktop command line syntax.

To use script arguments, add the `-scriptargs` parameter and specify the arguments. For example:

```
ansysedt.exe -scriptargs "hello there"
```

The command line parameter following `-scriptargs` is passed without modification as a single string in the ScriptArgument python variable.

For more information about running a script from the command line, consult the HFSS 3D Layout help topic "Running Ansys Electronics Desktop from the Command Line".

Recording a Script

Electronics Desktop can record a script based on UI actions and save this script in Python (*.py) format.

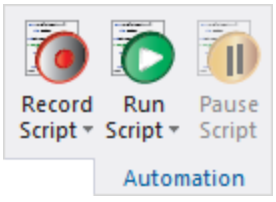
Scripts can be saved to an external file, or to the project.

> **Important:**
>
> When you record a script, every subsequent action you take is recorded. You must manually stop recording.

Recording a Script to File

To record a script to file:

1. Click **Tools** > **Record Script to File**, or select the **Automation** tab and click the **Record Script** icon:



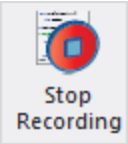
   A **Save As** file browser appears.

2. Navigate to the location where you want to save the script.
3. In the **File Name** field, type a name for the script file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

4. Click **Save**.

   The **Record Script** button transforms into a **Stop Recording** button, and Electronics Desktop begins recording your actions.



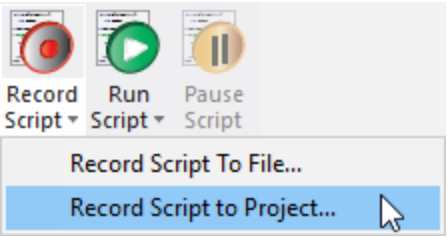
5. Perform the steps you want to record.
6. When you have finished recording the script, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

   The recorded script is saved to the folder you specified.

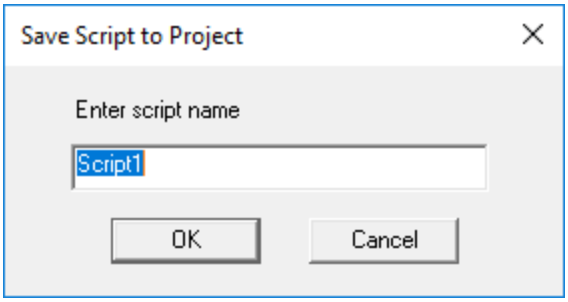
Recording a Script to a Project

To record a script to a project:

1. Click **Tools** > **Record Script to Project**, or select the **Automation** tab and use the **Record Script** drop-down menu to select **Record Script to Project**.



   The **Save Script to Project** dialog box appears:



2. Enter a name for the script in the text box, then click **OK**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

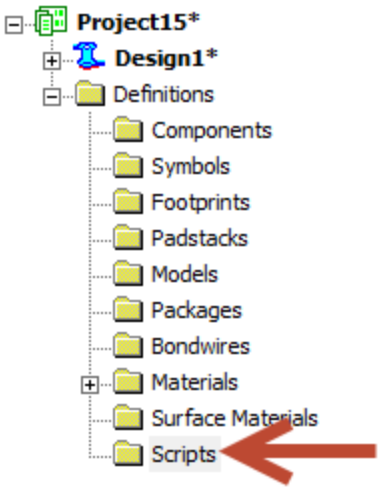
3. Perform the steps you want to record.
4. When you have finished, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

   The recorded script is saved to *scriptname.py* in the Scripts library and can be accessed from the Project Manager. See: Working with Project Scripts.

Working with Project Scripts

Scripts can be recorded to a project.

Once a script has been recorded to the project, you can manage it in the **Project Manager** from the **Definitions** folder:



Individual scripts appear in this folder. Right-click a script to edit or run it:

You can also run project scripts from the **Automation** tab by selecting **Run Script** > **Project Scripts** > [**Script Name**].

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

> **Note:**
>
> Project scripts are stored in the project scripts library. Refer to the topic "Managing Library Contents" for information on working with libraries.

Executing a Script from Within a Script

Electronics Desktop provides a script command that enables you to launch another script from within the script that is being executed.

In CPython:

```
oDesktop.RunScript(<ScriptName>)
```

If the full path to the script is not specified, Electronics Desktop searches for the specified script in the following locations, in order:

1. Personal Library Directory (PersonalLib)
2. User Library Directory (UserLib)
3. System Library Directory (SysLib)
4. Installation Directory

Each of the library directories can be specified in Electronics Desktop under **Tools** > **Options** > **General Options**, on the **Project Options** tab.

Electronics Desktop Scripting Conventions

A number of scripting conventions exist for Electronics Desktop regarding syntax, arguments, and numerical values.

Consult the following topics:

- Named Arguments
- Setting Numerical Values
- Layout Scripts and the Active Layer
- Scripts and Locked Layers

Named Arguments

Many Electronics Desktop script commands use named arguments. The names can appear in three ways:

1. Named data, where name precedes data.

   For example:

   ```
   ..., "SolveInside:=", true, ...
   ```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

2. Named Array, where name precedes array.

   For example:

   ```
   ..., "Attributes:=", [...],...
   ```

3. Named Array, where name is inside an array.

   For example:

   ```
   ..., ["NAME:Attributes",...],...
   ```

In the first and second examples, the name is formatted as `"<Name>:="`. This signals to Electronics Desktop that this is a name for the next argument in the script command. In the third example, the name is formatted as `"NAME:<name>"` and is the first element of the array.

The names are used both to identify what the data means to you and to inform Electronics Desktop which data is being given. The names must be included or the script will not play back correctly. However, if you are writing a script, you do not need to pass in every piece of data that the command can take. For example, if you are modifying a boundary, the script will be recorded to include every piece of data needed for the boundary, whether or not it was modified. If you are writing a script by hand, you can add only the data that changed and omit anything that you do not want to change. Electronics Desktop will use the names to determine which data you provided.

IronPython Example

When editing a port excitation, Electronics Desktop records the `Edit` command as follows in IronPython:

```
oModule.Edit("Port1",
  ["NAME:Port1",
    ["NAME:Properties",
    "PortSolver:=", "true",
    "Phase:=", "0deg",
    "Magnitude:=", "2mA",
    "Impedance:=", "50Ohm",
    "Theta:=", "0deg",
    "Phi:=", "0deg",
    "PostProcess:=", "false",
    "Renormalize:=", "50Ohm + 0i Ohm",
    "Deembed:=", "0mm",
    "RefToGround:=", "false"
    ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Type:=", "EdgePort",
  "IsGapSource:=", true,
  "UpperProbe:=", false,
  "LayoutObject:=", "Port1",
  "Pin:=", "",
  "ReferencePort:=", ""
  ]
)
```

If you only wish to change the magnitude, you can leave out the other data arguments when manually writing a script:

```
oModule.Edit("Port1",
  ["NAME:Port1",
    ["Magnitude:=", "1mA"]
  ]
)
```

Setting Numerical Values

For script arguments that expect a number, the following options are possible:

- Pass in the number directly.

  ```
  oModule.EditVoltage('Voltage1', ['NAME:Voltage1', 'Voltage:=',
  3.5])
  ```

- Pass in a string containing the number with units.

  ```
  oModule.EditVoltage('Voltage1', ['NAME:Voltage1', 'Voltage:=',
  '3.5V'])
  ```

- Pass in a variable name.

  ```
  var = 3.5

  oModule.EditVoltage('Voltage1', ['NAME:Voltage1', 'Voltage:=',
  var]
  ```

Layout Scripts and the Active Layer

A design's active layer is the layer that is used for object creation and placement during adding operations in the user interface. Adding operations include paste and placement of instances, as well as object creation. Usually there is an active layer, but it is not required and cannot be assumed. Adding operations are responsible for ensuring that the active layer exists and meets

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

any particular requirements (such as layer type) for the operation. Adding operations may change the active layer to a different layer that meets requirements. If the active layer is changed, Electronics Desktop generates an alert. If no layer is available to be active, the operation is not done.

The active layer is not used during script adding operations. Script adding operations are responsible for ensuring that the specified layer exists and meets the particular requirements (such as layer type) for the operation. If there is a problem with using the specified layer, the operation is not done. The active layer is always visible and selectable. These attributes are reset, if needed, when a layer is made active. The current active layer is indicated by a combo box display in the toolbar. The list for the combo box contains all layers that may be set active.

The active text style is related to the active layer. If there is no active layer, there is no active text style. Objects on the active layer have priority during snapping.

Scripts and Locked Layers

The locked attribute of a layer is defined to mean that you may not edit, delete, or add objects on the layer, either directly or with scripts (i.e., scripts run on layout or footprint definitions). This includes not being able to change properties of objects on the layer. Note, however, that parameter changes can alter objects on locked layers.

The locked attribute of a layer is configurable using script commands and is user-editable via the **Edit Layers Dialog** in the Layout Editor.

PyAEDT (Beta)

PyAEDT is a Python library that interacts with the Electronics Desktop API to make scripting simpler for the end user.

It supports:

- All 3D products (HFSS, Icepak, Maxwell 3D, and Q3D Extractor)
- 2D tools
- Ansys IcepakFEA
- EMIT
- Circuit Nexxim
- Twin Builder
- Layout tools (HFSS 3D Layout and EDB)

Additionally, it enables the end user to have a CPython interface with AEDT. Its class and method structures simplify operation for the end user, enabling more Pythonic code while reusing information as much as possible across the various APIs.

Documentation for PyAEDT, including installation instructions, extensions, and example scripts can be found online at:

https://aedt.docs.pyansys.com/version/stable/

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Installing PyAEDT adds three items to the **Tools** > **Toolkit** > **PersonalLib** menu and to the **Automation** tab:

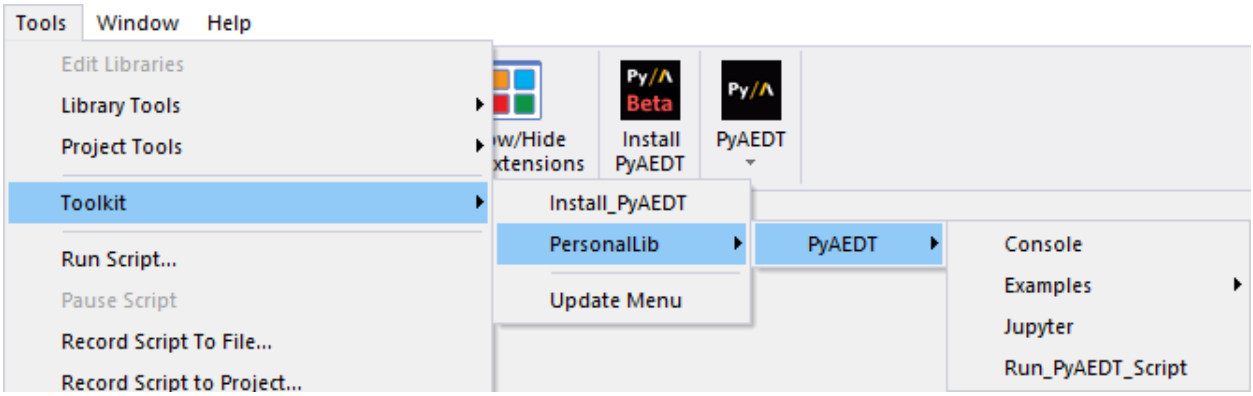
- **Console** – launches the PyAEDT console.
- **Jupyter Notebook** – launches Jupyter Notebook (a computational notebook) in an internet browser.
- **Run PyAEDT Script** – launches a file browser allowing you to select a Python script to run via PyAEDT.

To install PyAEDT:

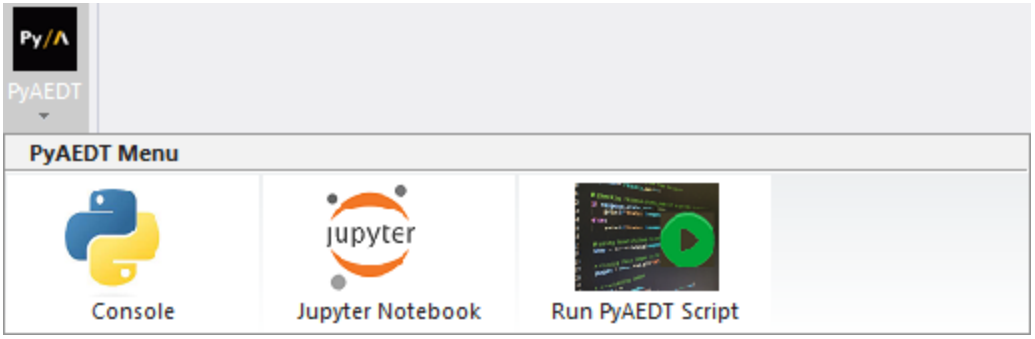
- From the **Automation** tab, click **Install PyAEDT**.

A web browser launches, and takes you to detailed installation instructions.

When installation is complete, the **Tools** > **Toolkit** >**PersonalLib** menu and the **Automation** tab update to display PyAEDT menu options:



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

2 - Object-Oriented Scripting

Object-oriented scripting enhances scripting in Electronics Desktop by allowing object-oriented access to retrieve or modify object properties. The primary gain is the ease with which properties of existing objects in a project or design can be read, modified, and set. This feature supports the use of less code to access object properties and enhances code readability by avoiding complex array inputs.

Object-Oriented Scripting Logic/Syntax

There are five basic functions in object-oriented scripting for retrieving and setting properties:

Function	Purpose
GetChildNames()	Lists child object names.
GetChildObjects()	Returns child object instances.
GetPropNames()	Lists available properties.
GetPropValue()	Retrieves a property's value.
SetPropValue()	Modifies a property's value.

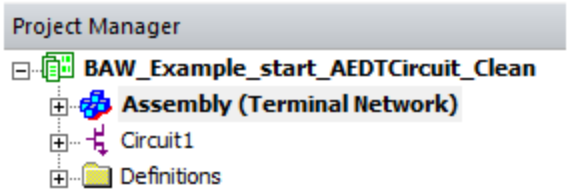
Basic Functions Overview

This section describes a typical object-oriented scripting workflow.

At a high level, use GetChildNames() to determine what object instances exist for a given object.

For example, the Electronics Desktop project below contains two designs ('Assembly' and 'Circuit1'):

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Open the Command Window and define the Project Object (oProject) and the Design Object (oDesign):

```
oProject = oDesktop.GetActiveProject()
oDesign = oProject.GetActiveDesign()
```

Once the objects have been defined, use GetChildNames() to learn what object instances exist for each. Per the example above, the Child Names of oProject are the names of the designs included in the project:

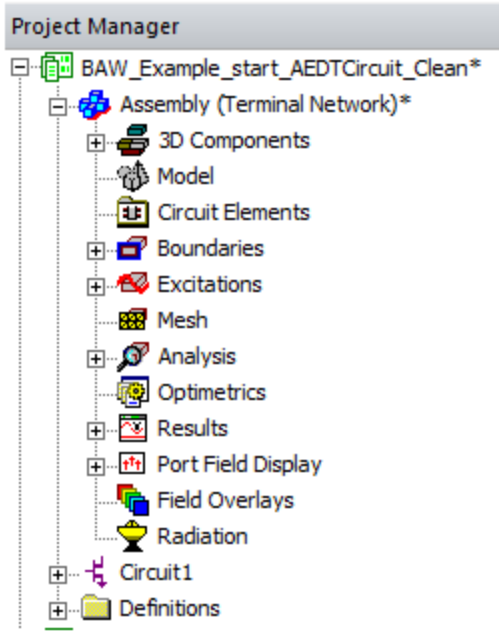
```
>>> oProject.GetChildNames()
['Assembly','Circuit1']
```

As another example, retrieve the names of the Object Instances available in oDesign to see the various objects associated with a Design setup:

```
>>> oDesign.GetChildNames()
['Boundaries', 'Excitations', 'Circuit Elements', 'Model', 'Mesh', 'Analysis', 'Optimetrics',
'Port Field Display', 'Field Overlays', 'Radiation', 'Results', '3D Modeler']
```

These names are roughly equivalent to the items nested beneath the design in the Project Manager:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

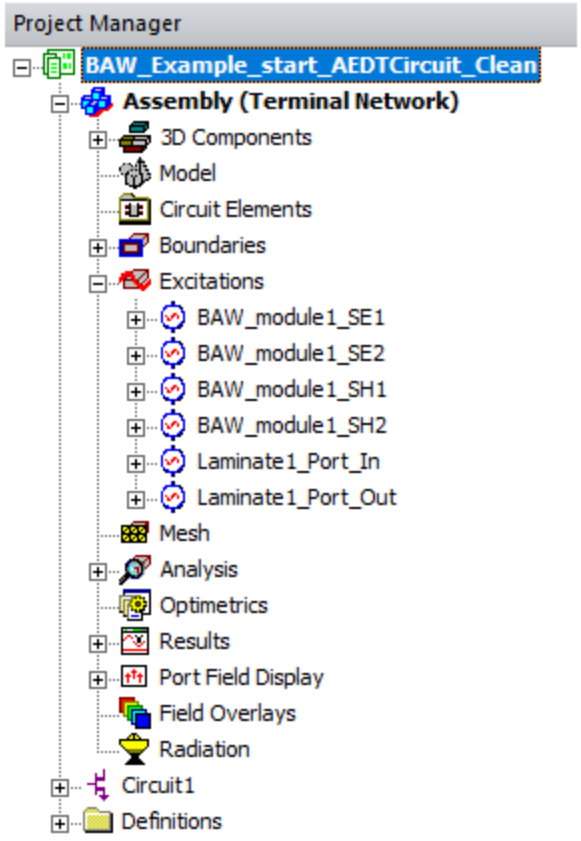


Any object in the **Project Manager** that has the '+' symbol for expansion is populated with child objects that can be queried. Once again, GetChildObject() can instance these objects. For example, set an instance for the Excitations:

```
oExcitations = oDesign.GetChildObject('Excitations')
```

The object oExcitations is now defined to be the oDesign Child Object 'Excitations'. Expanding **Excitations** in the **Project Manager**, there are six ports defined:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



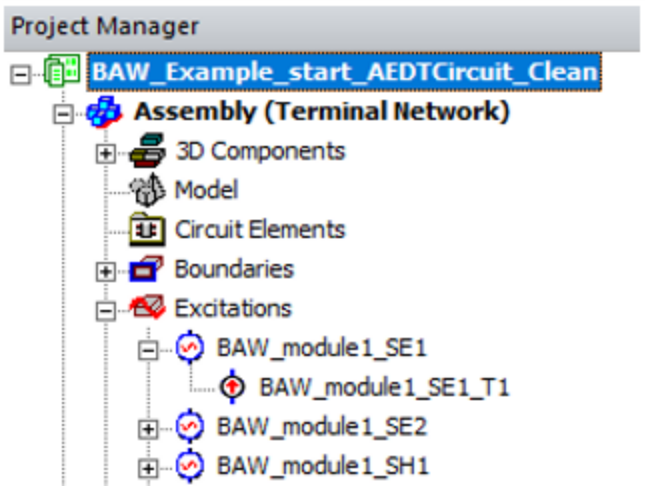
Those ports are outputted using GetChildNames():

```
>>> oExcitations.GetChildNames()
['Laminate1_Port_In', 'Laminate1_Port_Out', 'BAW_module1_SE1', 'BAW_module1_SE2', 'BAW_mod-
ule1_SH1', 'BAW_module1_SH2']
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

There is a clear logic, as the children of the oExcitations object are the ports that have been defined in HFSS. Exploring the **Project Manager** window can help you to conceive and retrieve desired information.

Expanding the first port reveals its own child object, its terminal:



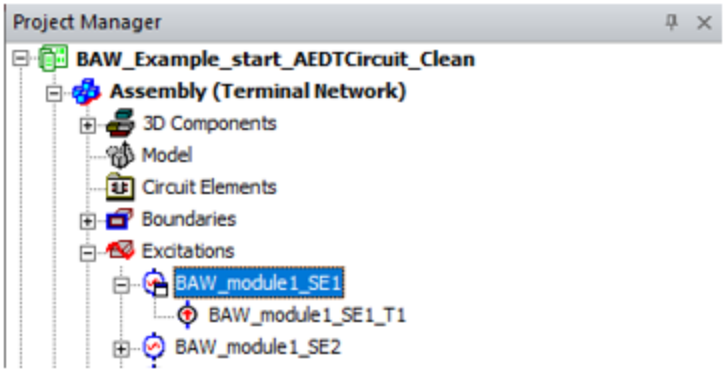
Through scripting, first define the Port Object (in this example, oPort) using the 'GetChildObject()' command for the first port, 'BAW_module1_SE1.' Then determine its Object Child Name, the terminal definition:

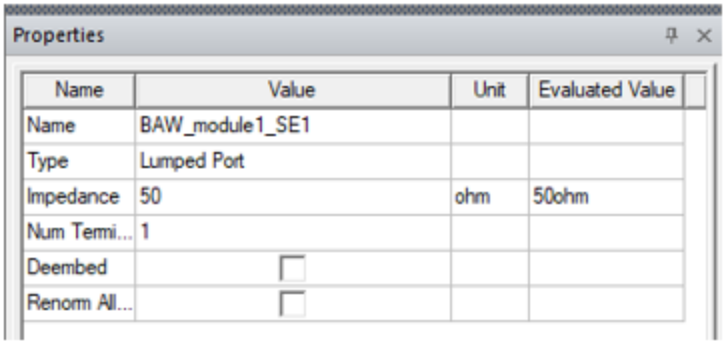
```
>>> oPort = oExcitations.GetChildObject('BAW_module1_SE1')
>>> oPort.GetChildNames()
['BAW_module1_SE1_T1']
```

As the use and logic for GetChildNames() and GetChildObject() have been demonstrated, you can now explore the properties of each of these objects, if they exist. The function to determine what properties exist is GetPropNames(). Use this to determine what properties exist to be retrieved or modified for a given object. The properties available are readily identifiable in the **Properties** window,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

which by default is located beneath the **Project Manager** window. For example, if you select a given port object, 'BAW_module1_SE1', the Properties window populates as shown:





If you execute the GetPropNames() function on the previously defined object, oPort, you see the same Property Names as available in the **Properties** window:

```
>>> oPort.GetPropNames()
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
['Name', 'Type', 'Impedance', 'Num Terminals', 'Deembed', 'Renorm All Terminals']
```

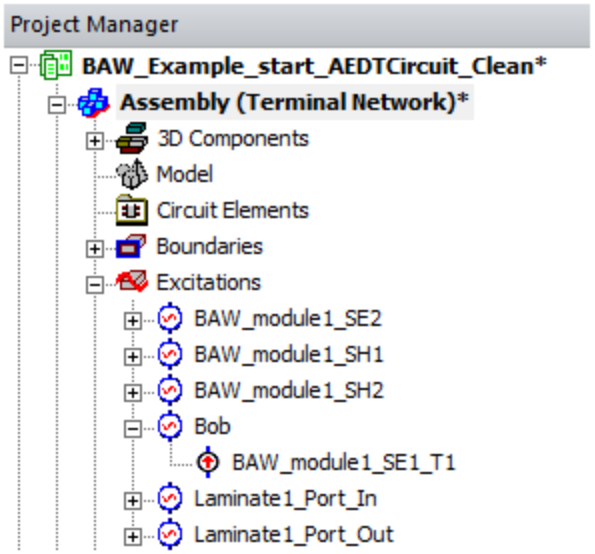
Once you identify the desired object and know the desired property, you can access the value via GetPropValue(). For example, if you want to retrieve the name of the object oPort:

```
>>> oPort.GetPropValue('Name')
'BAW_module1_SE1'
```

To change the value of the property, use the SetPropValue() function. The arguments for this function are (*'Property Name'*,*'New Value'*). For example, to change the name of the port to 'Bob':

```
>>> oPort.SetPropValue('Name', 'Bob')
True
```

This function returns a Boolean 'True' if successful. The Project Manager window updates accordingly:



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This approach for retrieving and setting properties is general and can be used for many aspects of an Electronics Desktop simulation. This object-oriented method of property identification and modification operates only on existing objects. Object-oriented scripting cannot create new instances; you must revert to the functions in a given module to do that. Not all children of a given object may be accessible via the GetChildNames() command. For help with specific objects, reference details in the Scripting Help or reach out to an Application Engineer.

Body Properties and Modification

The following example shows how to retrieve the properties of a body in the model, in this case a Region object. Once you identify the desired property, you can modify it as needed.

```
>>> oModel = oDesign.GetChildObject('3D Modeler')
>>> oModel.GetChildNames()
['RadBox_Region_1']
>>> oRegion = oModel.GetChildObject('RadBox_Region_1')
>>> oRegion.GetPropNames()
['Name', 'Material', 'Solve Inside', 'Orientation', 'Orientation/Choices', 'Model', 'Group', 'Display
Wireframe', 'Material Appearance', 'Color', 'Color/Red', 'Color/Green', 'Color/Blue', 'Transparent']
```

Retrieving Variables

Retrieving defined variables in a design or project is a common effort for automation. There are two types of variables: design and project. Project variables are preceded with a '$' symbol and are retrieved in the project object as it is globally defined to all designs. Design variables do not have any preceding symbols and are retrieved in the design object as their scope is limited to a given design. The following example demonstrates the retrieval of project variable names and values, and then design variable names and values:

```
>>> oProjVar = oProject.GetChildObject("Variables")
>>> oProjVar.GetPropNames()
['$test']
>>> oProjVar.GetPropValue('$test')
'0'
>>> oDesVar = oDesign.GetChildObject("Variables")
>>> oDesVar.GetPropNames()
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
['test']
>>> oDesVar.GetPropValue('test')
'0'
```

Retrieving Datasets and Values

The GetChildObject, GetChildTypes and GetChildNames functions operate on the oDesktop objects. This allows you to retrieve and view datasets and values. The dataset script wrapper store all values internally in SI units, and converts them back to user-supplied units when you request non-SI property values.For example, if you assigned a dataset to the example OptimTee project in HFSS, you could use these functions in the command window:

```
>>> oDesktop.GetChildTypes()
['Projects']
>>> oDesktop.GetChildNames()
['OptimTee']

>>> arrProjectNames = oDesktop.GetChildNames()
>>> tp = oDesktop.GetChildObject('OptimTee')
>>> tp.GetChildTypes()
['Design', 'Project Data']
>>>tp.GetChildNames('Project Data')
['Variables', 'Materials', 'Surface Materials', 'Datasets']
>>> ds = tp.GetChildObject('datasets')
>>>ds.GetChildNames()
['$ds1']
>>> ds1=ds.GetChildObject('$ds1')
>>> ds1.GetPropValue('[:,:]')
[[1.0, 4.0], [2.0, 5.0], [3.0, 6.0]]
>>> ds1.GetPropSIValue()
[[1.0, 4.0], [2.0, 5.0], [3.0, 6.0]]
>>> ds1.DimUnits

>>> ds1.DimUnits = ['mm','mm']
>>> ds1.DimUnits
['mm', 'mm']
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
>>> ds1.GetPropSIValue()
[[0.001, 0.0040000000000000001], [0.002, 0.0050000000000000001], [0.0030000000000000001,
0.0060000000000000001]]
>>> ds1.GetPropValue('[:,:]')
[[1.0, 4.0], [2.0, 5.0], [3.0, 6.0]]
```

GetSolutionData API

Many users want to use scripts to extract solution data from Ansys Electronics Desktop for custom post-processing. Scripting includes a new method to do this without having to export data to a file and then re-import it for use in a script. The new function is accessible via the ReportSetup module. The function call is GetSolutionDataPerVariation(). A code snippet to extract Terminal S Parameter data is shown:

```
oModule = oDesign.GetModule("ReportSetup")
Results = oModule.GetSolutionDataPerVariation("Terminal Solution Data", "Setup1 : Sweep1:",
    [
        "Domain:=", "Sweep"
    ],
    [
        "Freq:=", ["All"]
    ],
    [
        "dB(St(Terminal_1,Terminal_1))"
    ]
##  Get Dependent and Independent Variable data for Nominal Variation
NominalData = Results[0]
##  Get Independent Variable Data
##  For second arguement, if pass True then data is in SI Units
##      if pass False then data is in default scale units instead of SI
SweepValues = NominalData.GetSweepValues("Freq", True)
##  Get Dependent Variable DataValues
##  Note:  Can pass any 'Y Component' Name
DataValues = NominalData.GetRealDataValues("dB(St(Terminal_1))")
```

The above code shows how to extract the dependent and independent data to variables for easy manipulation. For more information on other functions available for this, see GetSolutionDataPerVariation.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Summary

Scripting has been advancing in Ansys Electronics Desktop to better allow you to customize and automate their repetitive or complex simulations. The ability to easily retrieve and set property values via the object-oriented scripting allows for ease or both writing and reading. The ability to extract solution data within a script execution is a new functionality that markedly enables more advanced post-processing.

Object-oriented property scripting presents an easy to use, intuitive and object-oriented representation of the data model. The framework supports query of objects and their properties including the edits of the data model in an object-oriented fashion. With this scripting framework, data exposure is intuitive and provides maximum coverage.

Each exposed script object supports the following COM functions:

- GetName
  - Return name of the object as text string
  - *e.g.*, name of a design, solve setup, boundary, etc
- GetChildTypes
  - An object can have different types of children.
  - Return array of text string. Can be empty if the object's children are NOT categorized into different types.

  For example, a design object has three children types. The following examples show how the commands run in the **Tools** > **Open Command Window** for IronPython.

```
>>> design.GetChildTypes()
['Module', 'Editor', 'Design Data']
```

- GetChildNames
  - Input: [String – Type]. Default = "Module" and "Editor" for design script object. 'All' for other script objects.
  - Return an array of immediate children's names, of a given type if specified

  For example, an IcepakFEA design object has these children:

```
>>> design.GetChildNames()
['Boundaries', 'Excitations', 'Optimetrics', 'Results', '3D Modeler']
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Four of the children are of “Module” type:

```
>>> design.GetChildNames("module")
['Boundaries', 'Excitations', 'Optimetrics', 'Results']
```

- GetChildObject
  - Input: String -- Object path. The path may include multiple generations.
  - Return the child object if found

For example,

```
>>> d = project.GetChildObject("hfss")
>>> d.GetChildObject("3d modeler").GetChildNames()
['Box1', 'Box1_1', 'Box1_1_1']
>>> project.GetChildObject("hfss/3d modeler").GetChildNames()
['Box1', 'Box1_1', 'Box1_1_1']
```

- GetPropNames
  - Input: [BOOL - IncludeReadOnly] -- default to true
  - Return an array of the object’s properties

For example,

```
>>> geom = project.GetChildObject("hfss/3d modeler").GetChildObject("Box1")
>>> geom.GetPropNames()
['Name', 'Material', 'Material/SIValue', 'Material/EvaluatedValue', 'Solve Inside', 'Ori-
entation', 'Orientation/Choices', 'Model', 'Group', 'Display Wireframe', 'Material Appear-
ance', 'Color', 'Color/Red', 'Color/Green', 'Color/Blue', 'Transparent']
```

- GetPropValue
  - Input: String – Property Path. The path may include multiple generations.
  - Return the property value as VARIANT

For example,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
>>> geom.GetPropValue("material")
'"vacuum"'
>>> geom.GetPropValue("xsize")
'3mm'
>>> op.GetPropValue("attach to original object")
False
```

- SetPropValue
  - Input: String – Property Path. The path may include multiple generations.
  - Input: String – Data. New value of the property.
  - Return – True if property data is updated successfully. False if failed to assign the new value.

  For example,

```
>>> geom.SetPropValue("model", False)
>>> boxcmd.SetPropValue("ysize", "4mm")
```

- GetPropEvaluatedValue (<*PropName*>)

  For example,

```
oVar = oDesign.GetChildObject(" Variables/var")
oVar.GetPropEvaluatedValue()
```

- GetPropSIValue (<*PropName*>

  For example,

```
oCreateBox = oDesign.GetChildObject("3D Modeler/Box1/CreateBox:1")
```

```
oCreateBox.GetPropValue("xSize")
```

returns `"length / 2"`

```
oCreateBox.GetPropEvaluatedValue("xSize")
```

returns `'0.4mm'`

```
oCreateBox.GetPropSIValue("xSize")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

returns `'0.0004'`

Additional Details Specific to AEDT Solvers

“Variables” and “Design Settings” are exposed as “Design Data” type children of a design script object.

The following “Module” types are exposed as “Module” type children of a design script object.

HFSS

- Boundaries, Excitations, Circuit Elements, Hybrid Regions, Analysis, Radiation, Field Overlays, Optimetrics, Results

HFSS 3D Layout

- Boundaries, Excitations, Circuit Elements, Analysis, Radiation, Field Overlays, Optimetrics, Results

Maxwell 3D/2D

- Boundaries, Excitations, Analysis, Field Overlays, Optimetrics, Results

RMxprt

- Analysis, Field Overlays, Optimetrics, Results

Q3D

- Boundaries, Nets, Analysis, Optimetrics

2D Extractor

- Boundaries, Conductors, Analysis, Field Overlays, Optimetrics

Icepak

- Thermal, Monitor, Mesh, Analysis, Field Overlays, Optimetrics, Results

IcepakFEA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- Boundaries, Excitations, Analysis, Field Overlays, Optimetrics, Results

Circuit

- Optimetrics, Results

EMIT

- Coupling

Simplorer/Twin Builder

- Analysis, Optimetrics, Results

Additional Details on Boundaries/Excitations

Each design type presents its boundaries/excitations data in the project tree as different groups. For example, an HFSS design has Boundaries, Excitations, Circuit Elements and Hybrid Regions while a Icepak design has just a “Thermal” project tree folder.

These module script objects do not have properties

```
>>> project.GetChildObject("icepak/thermal").GetPropNames()
[]
```

GetChildTypes of these module script objects returns the types of its immediate children

```
>>> d = p.GetChildObject("q2d")
>>> d.GetChildObject("conductors").GetChildTypes()
['NonIdealGround', 'SignalLine']
```

GetChildNames of these module object returns its immediate children

```
>>> p.GetChildObject("icepak/thermal").GetChildNames()
['Source1', 'Resistance1', 'ConductingPlate1', 'Source2', 'Resistance2', 'ConductingPlate2',
'Source3', 'Resistance3', 'ConductingPlate3']
```

GetChildNames can be invoked with a “type” and the returns will be filtered by that given type.

```
>>> p.GetChildObject("icepak/thermal").GetChildNames("resistance")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
['Resistance1', 'Resistance2', 'Resistance3']
```

Children of a module object are scriptable objects and have properties.

```
>>> port = p.GetChildObject("hfss/excitations/1")
>>> port.GetPropNames(False)
['Name', 'Deembed', 'Deembed Dist', 'Renorm All Terminals']
```

You can query/edit these properties

```
>>> port.GetPropValue("deembed")
False
>>> port.SetPropValue("deembed", True)
True
>>> port.GetPropValue("deembed")
True
```

A boundary/excitation script object can also have children. For example, HFSS terminal is a child of its port. Q3D source/sink can be children of a net.

```
>>> port.GetChildNames()
['Box1_T1']
>>> port.GetChildTypes()
['Terminal']
>>> p.GetChildObject("q3d/nets/s2").GetChildNames()
['Source2', 'Sink2']
>>> p.GetChildObject("q3d/nets/s2").GetChildTypes()
['Sink', 'Source']
```

3D Component Encapsulation

These script interfaces are compliant with encapsulation. For example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- Design.GetChildObject(“boundaries”).GetChildNames() will not return component boundaries
- SetPropValues of component excitations can only be used to edit post processing settings such as 'Deembed', 'Deembed Dist' of a HFSS port.

Additional Details on Solve Setup

All solve setups are children of the “Analysis” script object. This parent script object is also of the type “Module”.

```
>>> d = oDesktop.GetActiveProject().GetChildObject("hfss")
>>> d.GetChildNames()
['Boundaries', 'Excitations', 'Hybrid Regions', 'Circuit Elements', 'Analysis', 'Optimetrics',
'RadField', 'Results', '3D Modeler']
>>> setups = d.GetChildObject("analysis")
```

This module script object has no property

```
>>> setups.GetPropNames()
[]
```

The children of this module script object in a 3D design is not categorized into different types because the solve setup type is one-to-one to the solution type of a 3D design.

```
>>> setups.GetChildTypes()
[]
```

The children of this module script object in a 3D Layout and Simplorer/Twin Builder design is categorized into different solve setup types, such as “Transient”, “AC” and “DC” in a Simplorer/TwinBuilder design and “HFSS” and “SIwave” in a 3D layout design.

A solve setup script object can also have children. Children are typically frequency sweeps.

```
>>> setup.GetChildNames()
['Sweep', 'Sweep1', 'Sweep2']
>>> setup.GetChildTypes()
['Discrete', 'Interpolating']
>>> sweep1 = setup.GetChildObject("sweep")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Object-Oriented Scripting for Editors

Editors can be accessed via object-oriented scripting.

"3D Modeler" is a child of MCAD design (HFSS, Maxwell3D/2D, Q3D, 2D Extractor, Icepak and IcepakFEA).

"Machine" is a child of RMxprt design.

"SchematicEditor" is a child of circuit design (Circuit, Simplorer/Twin Builder and Maxwell Circuit)

Schematic Editor

Schematic has 3 types of objects: Component, Port and Net. Data of these types can be accessed via object-oriented scripting.

Invoking GetChildNames without a “type” returns objects of the default type (Component).

> **Note:**
>
> A port can be of component type and port type. Invoke GetChildObject when “type” to retrieve the object of the desired type.

Object-Oriented Scripting for Materials

This section describes material properties and how to access and modify them.

Because materials are globally defined, the objects are children of the project, oProject, as shown below:

```
>>> oProject = oDesktop.GetActiveProject()
>>> oMaterials = oProject.GetChildObject('Materials')
>>> oMaterials.GetChildNames()
['vacuum', 'Cap_Mat', 'Outline_Mat', 'SolderMask_Mat', 'copper', 'pec']
```

Materials are child objects of the active project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

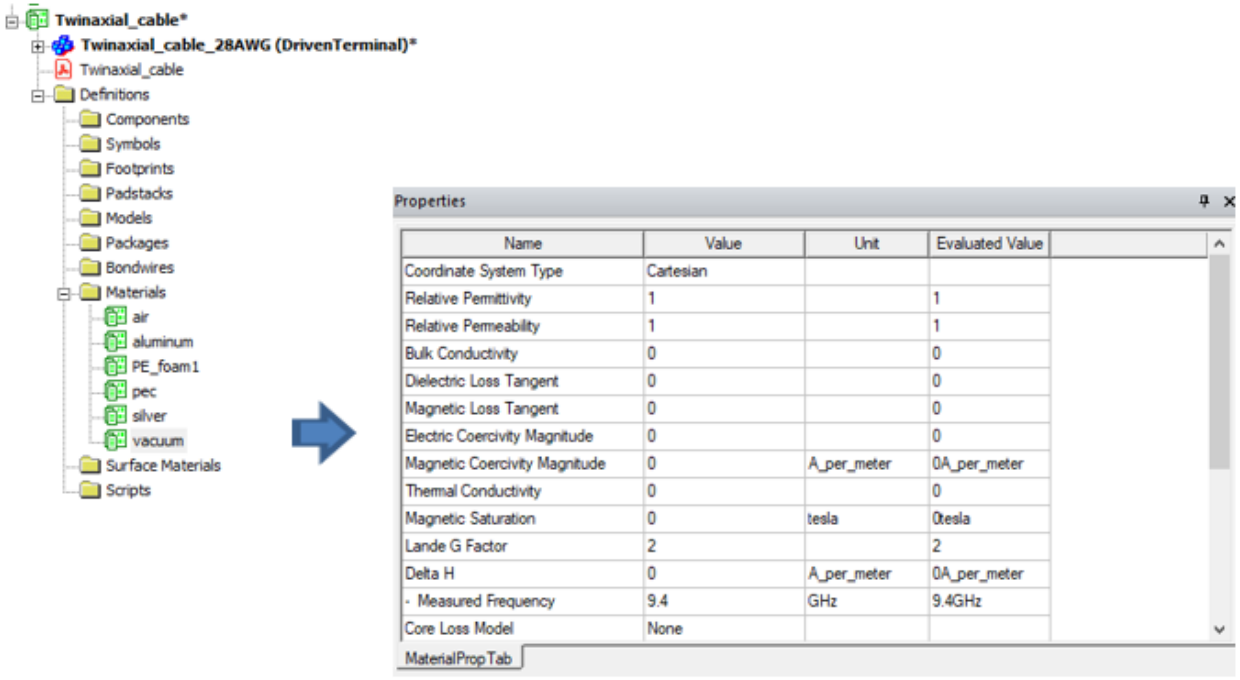
In the command window, execute GetPropNames() for a specified material as follows:

```
>>> omats = oDesktop.GetActiveProject().GetChildObject('Materials')

>>> omat = omats.GetChildObject('vacuum')

>>> omat.GetPropNames()

['Coordinate System Type', 'Coordinate System Type/Choices', 'Relative Permeability Type',
'Relative Permeability Type/Choices', 'Relative Permeability', 'Relative Per-
meability/SIValue', 'Relative Permeability/EvaluatedValue', 'Bulk Conductivity Type', 'Bulk
Conductivity Type/Choices', 'Bulk Conductivity', 'Bulk Conductivity/SIValue', 'Bulk Con-
ductivity/EvaluatedValue', 'Magnetic Coercivity Type', 'Magnetic Coercivity Magnitude', 'Mag-
netic Coercivity Magnitude/SIValue', 'Magnetic Coercivity Magnitude/EvaluatedValue',
'Composition', 'Composition/Choices', "Young's Modulus Type", "Young's Modulus Type/Choices",
"Young's Modulus", "Young's Modulus/SIValue", "Young's Modulus/EvaluatedValue", "Poisson's
Ratio Type", "Poisson's Ratio Type/Choices", "Poisson's Ratio", "Poisson's Ratio/SIValue",
"Poisson's Ratio/EvaluatedValue"]
```

All materials with a Project Definition or assigned to an object in the project are accessible.

Supported material properties are shown in the Properties window for the material item.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.



Name	Value	Unit	Evaluated Value
Coordinate System Type	Cartesian		
Relative Permittivity	1		1
Relative Permeability	1		1
Bulk Conductivity	0		0
Dielectric Loss Tangent	0		0
Magnetic Loss Tangent	0		0
Electric Coercivity Magnitude	0		0
Magnetic Coercivity Magnitude	0	A_per_meter	0A_per_meter
Thermal Conductivity	0		0
Magnetic Saturation	0	tesla	0tesla
Lande G Factor	2		2
Delta H	0	A_per_meter	0A_per_meter
- Measured Frequency	9.4	GHz	9.4GHz
Core Loss Model	None		

Some material properties, such as Relative Permittivity, may have values assigned as BH Curves or Tensors, as described in the Assigning Materials section of the online help.

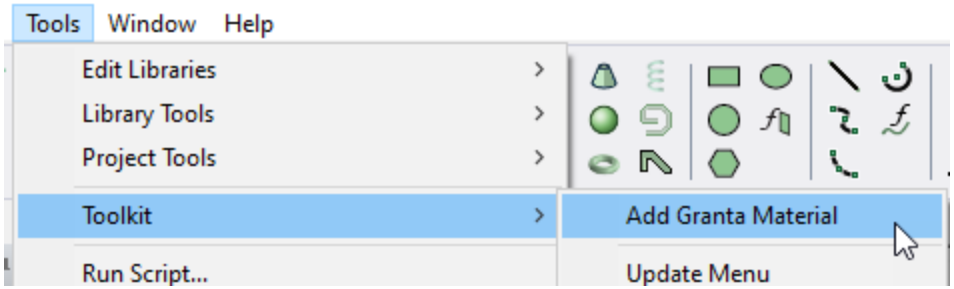
With this feature enabled you can then:

- Get/Set Simple material property
- Get/Set Anisotropic material property
- Get/Set Nonlinear material property

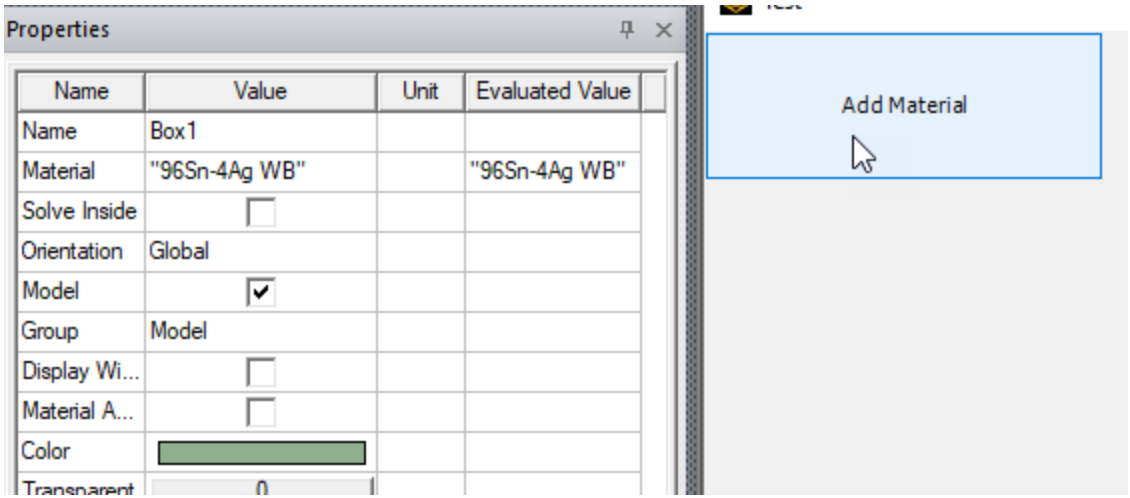
Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- Get/Set Vector material property
    - Components hide/shown as needed
- Get/Set Tensor material property
- Get/Set Choice material property
- AddDefinitionFromBlock
- AddDefiniitonFromLibFile
- GetExtendedDefinitionObect

A new Toolkit allows you to select materials from the Granta materials gateway, such that project materials will automatically be added when you select a material from the gateway, and that the gateway itself is easily accessed from the materials.



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



What is not supported:

- Change of property type
- Custom material property, due to its complexity

Example Change to Material Property Type:

```
>>>> omat.GetPropValue("Relative Permeability Type/Choices")
['Simple', 'Anisotropic', 'Tensor', 'Nonlinear']
>>>> omat.GetPropValue("Relative Permeability Type")
'Nonlinear'
>>> omat.SetPropValue("Relative Permeability Type", "Simple")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
True
>>> omat.GetPropValue("Relative Permeability Type")
'Simple'
>>> omat.SetPropValue("Relative Permeability", 10)
True
```

Example Change to Vector Component Value:

```
>>> omat.GetPropValue("Magnetic Coercivity Magnitude")
'0A_per_meter'
>>> omat.SetPropValue("Magnetic Coercivity Magnitude", "-1A_per_meter")
True
>>> omat.GetPropNames()
['Coordinate System Type', 'Coordinate System Type/Choices', 'Relative Permeability Type',
'Relative Permeability Type/Choices', 'Relative Permeability', 'Relative Per-
meability/SIValue', 'Relative Permeability/EvaluatedValue', 'Bulk Conductivity Type', 'Bulk
Conductivity Type/Choices', 'Bulk Conductivity', 'Bulk Conductivity/SIValue', 'Bulk Con-
ductivity/EvaluatedValue', 'Magnetic Coercivity Type', 'Magnetic Coercivity Magnitude', 'Mag-
netic Coercivity Magnitude/SIValue', 'Magnetic Coercivity Magnitude/EvaluatedValue', 'Magnetic
Coercivity Components', 'Magnetic Coercivity Components/Component1', 'Magnetic Coercivity Com-
ponents/Component2', 'Magnetic Coercivity Components/Component3', 'Composition', 'Com-
position/Choices', '- Stacking Factor Type', '- Stacking Factor', '- Stacking Factor/SIValue',
'- Stacking Factor/EvaluatedValue', '- Stacking Direction', '- Stacking Direction/Choices',
"Young's Modulus Type", "Young's Modulus Type/Choices", "Young's Modulus", "Young's Mod-
ulus/SIValue", "Young's Modulus/EvaluatedValue", "Poisson's Ratio Type", "Poisson's Ratio
Type/Choices", "Poisson's Ratio", "Poisson's Ratio/SIValue", "Poisson's Ratio/EvaluatedValue"]
>>> omat.SetPropValue("Magnetic Coercivity Components/Component2", 2)
True
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
>>> omat.GetPropValue("Magnetic Coercivity Components")

['Component1:=', '2', 'Component2:=', '2', 'Component3:=', '0']
```

Example Change to Choice Property Value:

```
>>> omat.GetPropValue("Composition/Choices")

['Solid', 'Lamination', 'Litz Wire']

>>> omat.SetPropValue("Composition", "Lamination")

True

>>> omat.GetPropNames()

['Coordinate System Type', 'Coordinate System Type/Choices', 'Relative Permeability Type',
'Relative Permeability Type/Choices', 'Relative Permeability', 'Relative Per-
meability/SIValue', 'Relative Permeability/EvaluatedValue', 'Bulk Conductivity Type', 'Bulk
Conductivity Type/Choices', 'Bulk Conductivity', 'Bulk Conductivity/SIValue', 'Bulk Con-
ductivity/EvaluatedValue', 'Magnetic Coercivity Type', 'Magnetic Coercivity Magnitude', 'Mag-
netic Coercivity Magnitude/SIValue', 'Magnetic Coercivity Magnitude/EvaluatedValue', 'Magnetic
Coercivity Components', 'Magnetic Coercivity Components/Component1', 'Magnetic Coercivity Com-
ponents/Component2', 'Magnetic Coercivity Components/Component3', 'Composition', 'Com-
position/Choices', '- Stacking Factor Type', '- Stacking Factor', '- Stacking Factor/SIValue',
'- Stacking Factor/EvaluatedValue', '- Stacking Direction', '- Stacking Direction/Choices',
"Young's Modulus Type", "Young's Modulus Type/Choices", "Young's Modulus", "Young's Mod-
ulus/SIValue", "Young's Modulus/EvaluatedValue", "Poisson's Ratio Type", "Poisson's Ratio
Type/Choices", "Poisson's Ratio", "Poisson's Ratio/SIValue", "Poisson's Ratio/EvaluatedValue"]

>>> omat.SetPropValue("Magnetic Coercivity Components/Component2", 2)

True

>>> omat.GetPropValue("Magnetic Coercivity Components")

['Component1:=', '2', 'Component2:=', '2', 'Component3:=', '0']
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

Property Script Commands

Property script commands allow you to navigate through all objects and properties in a project. You can get and set all properties for all objects in the Project tree with simple data types.

Property Object is the base class defined for all script objects that support the properties Get and Set.

`GetName()`

- Returns the name of the object.

`GetChildTypes()`

- An object may have different types of children. For example, a design may have variables, modules, and editors.
- Returns an array of text strings; may be empty if the children are not divided into different types.

`GetChildNames(<type>)`

- <type> – Child type name. By default, returns all children names for all types.
- Returns an array of immediate children names, belonging to a type if specified.

`GetChildObject(<objPath>)`

- <objPath> – A child object path; can contain multiple generations (for example, designObject/moduleObject/SetupObject).
- Returns a child property object if the object is found.

`GetPropNames(<bIncludeReadOnly>)`

- <bIncludeReadOnly) – Optional; defaults to true. True includes read-only properties; False excludes read-only properties.
- Returns an array of the object's property names.

`GetPropValue(<propertyPath>)`

- <propertyPath> – The property's path; may be a child object's path appended with a property name (for example, TeeModel/Offset/SIValue).
- Returns the property value if found. Otherwise causes script error.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

`SetPropValue(<propertyPath>, <data>)`

- <propertyPath> – The property's path; may be a child object's path appended with a property name (for example, TeeModel/Offset/SIValue).
- <data> – New data; type depends on property type.
- Returns True if updated successfully; False if new data is invalid.

For a detailed summary of how Property script commands are used in a range of contexts, including Variable objects, see: Object Script Property Function Summary. Additional examples for these commands are listed under Project Objects, Design Objects, 3D Modeler, Optimetrics, Radiation Module and Reporter.

> **Note:**
>
> Older property commands should be executed by the oProject object.
>
> ```
> oProject = oDesktop.SetActiveProject("Project1")
>
> oProject.CommandName <args>
> ```

Some of the topics covered in this chapter are as follows:

Object Script Property Function Summary

Conventions Used in this Chapter

Callback Scripting Using PropHost Object

ChangeProperty

GetArrayVariables

GetProperties

GetPropertyValue

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetVariables

GetVariableValue

SetPropertyValue

SetVariableValue

Example Use of Record Script and Edit Properties

Object Script Property Function Summary

Object Path

The Object path can be used to navigate through objects and properties in an Ansys EM project.

- An Object path consisted of one or multiple Object-ID-Nodes separated by "/" . In the HFSS-Optimtee example, the Optimization Max value of variable "offset" is represented by the path "TeeModel/offset/Optimization/max" or "TeeModel/Variable[offset]/Optimization[max]"
- Object-ID-Node; may exist in the following forms:
    - A simple object name or property name.
    - Type[Name] for object; Tab[name] for property.
    - Name[attr1="v1", attr2 = "v2", ...]. When more than one child object have the same name, use attributes to specify the difference.
    - ArrayName[index]. For example, in an Optimetric setup with multiple calculations, "Calculation[0]" could be used to identify the first calculation.
    - Name beginning with '@' character denoted as a property name, when an object has a child and property with the same name.

Property Object

The Property Object is the base class defined for all script object that support property Get & Set.

- GetName()
    - Returns the name of the object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- GetChildTypes()
    - An object may have different type of children. For example, a design may have variables, modules, and editors.
    - Returns array of text strings; may be empty if the children are *not* divided to different types.
- GetChildNames(<*type*>)
    - <*type*> – children type name; default returns all children names for all types.
    - Returns an array of immediate children names, belonging to the type if specified.
- GetChildObject(<*objPath*>)
    - <*objPath*> – A child object path. The path may include multiple generations, such as (designObject/moduleObj/SetupObject).
    - Returns a child property object if the object found.
- GetPropEvaluatedValue(<*propName*>)
    - Return the Evaluated-Value for Value-Property and Variable.
    - Return the Property-value as text string for other property types.
- GetPropSIValue(<*propName*>)
    - Return the SI-Value for Value-Property and Variable.
    - Return NAN for other property type if its value is cannot be converted to a double-floating point value.
- GetPropNames(<*bIncludeReadOnly*>);
    - <*bIncludeReadOnly*> – optional, default to true; True will include read-only properties, False will exclude read-only properties.
    - Returns an array of the object's property names.
- GetPropValue(<*propertyPath*>)
    - <*propertyPath*> – the property's full path. A property name or child object's path appended with a property name, like "TeeModel/Offset/SIValue"
    - Returns the property value if the property is foun; otherwise causes script error.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- SetPropValue(<*propertyPath*>, <*data*>)
  - <*propertyPath*> – the property's full path. A property name or child object's path appended with a property name, like "TeeModel/Offset/Value"
  - <*data*> – new data, type is dependent on property type.
  - Returns True if property data is updated successfully; False if the new data is invalid.

Project Object

Project Object inherited all functions defined in the Property Object. But it doesn't have property, GetPropValue() & SetPropValue() function can be used to set its child object's property.

- GetChildTypes() always return ["Design", "Variable"].
- GetChildNames(type)
  GetChildNames() & GetChildName("Design") will return all Design names of the project.
  GetChildNames("Variable") return all project variable names.
- GetChildObject(objPath)

  oDesign = oProject("TeeModel")

  oVariable = oProject.GetChildObject("VariableName")

  oReport = oProject.GetChildObject("TeeModel/Results/S Parameter Plot 1")

- GetPropNames(bIncludeReadOnly) always return empty array since the project has no property.
- GetPropValue(propertyPath)

  oProject.GetPropValue("TeeModel/offset") //get the offset variable value in the TeeModel Design
  oProject.GetPropValue("TeeModel/Results/S Parameter Plot 1/Display Type") // Get the report display type.

- SetPropValue(propertyPath, newValue)

  oProject.SetPropValue("TeeModel/offset", "2mm") //Set the offset variable value to "2mm" in the TeeModel Design
  oProject.SetPropValue("TeeModel/Results/S Parameter Plot 1/Display Type", "Data Table) // Set the report display type to data table.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Design Object

Design Object inherited all functions defined in the Property Object. But it doesn't have property, GetPropValue() & SetPropValue() function can be used to set its child object's property..

- GetChildTypes() always return ['Module', 'Editor', 'Variable'].
- GetChildNames(type)
  GetChildNames() will return modules & editor child names.
  GetChildNames(“Variable”) will return all variable names
  GetChildNames(“Module”) will return all module names that support property-object-script like ['Optimetrics', 'RadField', 'Results']
  GetChildNames(“Editor”) will return a 3D editor name for all 3D Designs
- GetChildObject()
  oVariable = oDesign.GetChildObject("VariableName")
  oReport = oDesign.GetChildObject("Results/S Parameter Plot 1")
  oRptModule = oDesign.GetChildObject("ReportSetup")
- GetPropNames(bIncludeReadOnly) always return empty array since the design has no property.
- GetPropValue()
  oDesign GetPropValue("offset/SIValue") //get the offset variable SI value in the DesignoDesign GetPropValue("offset/SIValue") //get the offset variable SI value in the Design
  oDesign.GetPropValue("Results/S Parameter Plot 1/Display Type") // Get the report display type
- SetPropValue()
  oDesign.SetPropValue("offset", "2mm") //Set the offset variable value to "2mm" in the Design
  oDesign.SetPropValue("Results/S Parameter Plot 1/Display Type", "Data Table) // Set the report display type to data table.

3D Modeler Object

GetChild commands returns the appropriate properties for modeler objects. For 3D Components and UDMs, these commands do not return parts, coordinate systems, plans, as top-level modeler children.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModeler = oDesktop.GetActiveProject().GetActiveDesign().GetChildObject("3D Modeler")
oModeler.GetChildNames()
oModeler.GetChildNames("ModelParts")
oModeler.GetChildNames("AllParts")
oModeler.GetChildNames("NonModelParts")
oModeler.GetChildNames("Planes")
oModeler.GetChildNames("CoordinateSystems")
```

Variable Object

Is a Property Object that has no child. It also provides quick function call to get/set it properties by adding functions with property name appended to Get_ & Set_ prefix. To find what functions it provided enter dir(oVar) the command window. It can accessed by the project or design object's GetChildObject(VariableName) function.

```
oProjVar = oProject.GetChildObject("$VarName")
oVar = oProject.GetChildObject("DesignName/VarName")
oVar = oDesign.GetChildObject("variableName")
oProject..GetChildNames("Variable") will return all project variable names.
oDesign.GetChildNames(Variable") will return all Design Variable names.
```

- GetChildTypes() always return empty array.
- GetChildNames() always return empty array , since variable has no child.
- GetChildObject(objPath) it has no child.
- GetPropNames(bIncludeReadOnly) ['EvaluatedValue', 'SIValue'] are read-only properties
    - Independent variable :['Value', 'EvaluatedValue', 'SIValue', 'Description', 'ReadOnly', 'Hidden', 'Sweep', 'Optimization/Included', 'Optimization/Min', 'Optimization/Max', 'Sensitivity/Included', 'Sensitivity/Min', 'Sensitivity/Max', 'Sensitivity/IDisp', 'Statistical', 'Statistical/Included', 'Tuning/Included', 'Tuning/Step', 'Tuning/Min', 'Tuning/Max'].
    - Dependent variable ['Value', 'EvaluatedValue', 'SIValue', 'Description', 'ReadOnly', 'Hidden', 'Sweep']
- GetPropValue(propName)
  oVar.GetPropValue() return the variable value as text string.
  oVar.GetPropValue("Value") return the variable value as text string.
  oVar.GetPropValue("SIValue") return the SI-value of variable as number.
  oVar.Get_SIValue()also return the SI value.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- SetPropValue(propName, newValue)
  oVar.SetPropValue(“Value", 888)
  oVar.SetPropValue("Sensitivity'/Included", True)
  oVar.SetPropValue("Sensitivity/Max”, '1.8pF'])
  oVar.Set_Sensitivity_Max('1.8pF') also works as last call.
  oVar.SetPropValue("Sensitivity", ['Min:=', '0.8pF', 'Max:=', '1.8pF'])
  //set multiple attributes at one call:
  oVar.SetPropValue("@", ['Value:=',288, 'Sensitivity', ['Included', True,'Min', '0.0']])
  oVar.SetPropValue("", ['Value:=',288, 'Sensitivity', ['Included', True,'Min', '0.0']])

Optimetrics Module Object:

Optimetrics Module Object inherited all functions defined in the Property Object. But it doesn't have property, GetPropValue() & SetPropValue() function can be used to set its child object's property..

- GetChildTypes() there are six type of children, they are ['OptiParametric', 'OptiOptimization', 'OptiSensitivity', 'OptiStatistical', 'OptiDesignExplorer', 'OptiDXDOE']. But the return array only included those that have setup defined, so it may be an empty array if no optimetrics setup is defined. The GetChildNames(type) function also recognized the type name without the prefix “Opti”.
- GetChildNames(type)
  GetChildNames() will return all setup for all types.
  GetChildNames(“OptiOptimization”) & GetChildNames(“Optimization”) will return all Optimization setup.
- GetChildObject()
  oParamSetup = oOptModule.GetChildObject('ParametricSetup1') get the
  oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')
- GetPropNames(bIncludeReadOnly) always return empty array since the it has no property.
- GetPropValue(propPath) may be used to get its child’s property value
  oOptModule.GetPropValue("OptimizationSetup1\Optimizer") get the optimizer name for OptimizationSetup1
- SetPropValue(propPath, newValue) may be used to set its child’s property value
  oOptModule.SetPropValue(ParametricSetup1\Enabled", False) //disable ParametricSetup1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Optimetrics Setup Object

This is a new Object inherited all functions defined in the Property Object. But it doesn't have child. It is accessible through its parents.

oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')

oOptSetup = oDesign.GetChildObject('Optimetrics\OptimizationSetup1')

oOptSetup = oProject.GetChildObject('TeeModel\Optimetrics\OptimizationSetup1')

- GetChildTypes() always return empty array.
- GetChildNames(type) always return empty array
- GetChildObject()
- GetPropNames(bIncludeReadOnly) will return the property names listed in the property window when the setup is selected.
- GetPropValue(propName)
  oOptSetup.GetPropValue("Optimizer") return the selected optimizer name.
  oOptSetup.GetPropValue("Optimizer/Choices") return all optimizer names.
- SetPropValue(propName, newValue)
  oOptSetup.SetPropValue("Optimizer", “NotAnOptimzerName”; will return false.
  oOptSetup.SetPropValue("Optimizer", “Quasi Newton”; return true, since “Quasi Newton” is one of the optimizer name returned as the Optimizer Choices.
- HasResult() return true if the setup is solved. Otherwise return false.
- Validate() return true if the setup is valid for analyze. Otherwise return false. Calling the SetPropValue() function to change the property may invalid the setup.

ReportSetup(Results) Module Object:

ReportSetup module Object inherited all functions defined in the Property Object. But it doesn't have property, GetPropValue() & SetPropValue() function can be used to get/set its child object's property..

- GetChildTypes() always empty array.
- GetChildNames(type)
  GetChildNames() return all report names

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- GetChildObject(objPath)
  oRpt = oRptModule.GetChildObject("S Parameter Plot 1") return the report property object
  oTrace = oRptModule.GetChildObject("S Parameter Plot 1/dB(S(Port1,Port1))") return the trace property object
  oAxisX = oRptModule.GetChildObject("S Parameter Plot 1/AxisX") return the axis X property object
- GetPropNames(bIncludeReadOnly) always return empty array since the it \has no property.
- GetPropValue()
  oRptModule.GetPropValue(“S Parameter Plot 1/Display Type”)
- SetPropValue()
  oRptModule.SetPropValue(“S Parameter Plot 1/Display Type”, “DataTable”)

ReportSetup(Results) Module Child Objects:

These are Property Objects. Its first level of child object is report. Report has trace, axis, header, Legend, and more children. Trace has curve as child etc.

Those child objects can be accessed by calling all levels of parent object’s GetChildObject(path) function.

oRpt = oRptModule.GetChildObject(reportName)

oRpt = oDesign.GetChildObject(“Results/reportName”)

oTrace = oRpt.GetChildObject(traceName)

oTrace = oRptModule.GetChildObject(ReportName/TraceName)

- GetChildTypes() always return empty array.
- GetChildNames() get the object’s child names. What will be returned will depended on the object instance.
- GetChildObject(objPath)
- GetPropNames(bIncludeReadOnly) will return the property names listed in the property window when the object is selected.
- GetPropValue(propName)
  oRpt.GetPropValue("Display Type") return the report’s display type.
  oOptSetup.GetPropValue("Display Type/Choices") return all optimizer names.
  oTrace.GetPropValue(“X Component”)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

- SetPropValue(propName, newValue)
  oTrace.SetPropValue(“Primary sweep”, “Freq”)

Radiation Module Object:

This inherited all functions defined in the Property Object. But it doesn't have property, GetPropValue() & SetPropValue() function can be used to set its child object's property.

- GetChildTypes() always return empty array, now its children
- GetChildNames(type)
  GetChildNames() return all setup names.
- GetChildObject(setupName) return the setup object as Property object.
  oOverlay= oRadModule.GetChildObject('Antenna Parameter Overlay1')
  oSphere = oRadModule.GetChildObject('Infinite Sphere1')
- GetPropNames() return empty array; it has no property.
- GetPropValue()
  oRadModule.GetPropValue('Line1/Num Points') //Get the the Line1 setups’ “Num Points” property value.
- SetPropValue()
  oRadModule.SetPropValue('Line1/Num Points', 100) ; Set the Line1 setups’ “Num Points” property to 100.

Radiation Module Child Objects:

These are Property Objects. It also provides quick function call to get/set its properties by adding functions with property name appended to Get_ & Set_ prefix. To find what functions it provides enter dir(oVar) the command window.

Those child objects can be access by call all levels of parent object’s GetChildObject(path) function.

oRadSetup = oRadModule.GetChildObject(setupName)

oRadSetup = oDesign.GetChildObject(RadField/setupName)

- GetChildTypes() always return empty array.
- GetChildNames() always return empty array , since Radiation setup has no child.
- GetChildObject(objPath) it has no child.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

- GetPropNames(bIncludeReadOnly) will return the property names listed in the property window when the setup is selected.
- GetPropValue(propName)
  oRadSetup.GetPropValue(”Num Points") return the line setup’s “Num Points” property value.
  oRadSetup.Get_NumPoints() will also get the same value.
- SetPropValue(propName, newValue)
  oRadSetup.SetPropValue('Num Points', 888)
  oRadSetup.Set_NumPoints(888)

Conventions Used in this Chapter

General Definitions:

Property	A single item that can be modified in the **Properties** window or in the modal **Properties** pop-up window.
<PropServer>	The item whose properties are being modified. This is usually a compound name, giving all information needed by the editor, design, or project in order to locate the item.
<PropTab>	Corresponds to one tab in the **Properties** window, the one under which properties are being edited.
<PropName>	The name of a single property.

The following tables list specific <PropServer> and <PropTab> values for different property types.

For Project Variables:

<PropServer>	"ProjectVariables"
<PropTab>	"ProjectVariableTab"

For Local Variables:

<PropServer>	"LocalVariables"
<PropTab>	"LocalVariableTab"

For Passed Parameters:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

<PropServer>	"Instance:<Name of Circuit Instance>"
<PropTab>	"PassedParameter Tab"

For Definition Parameters:

<PropServer>	"DefinitionParameters"
<PropTab>	"DefinitionParameters"

For Modules and Editors:

<PropServer>	<ModuleName>:<ItemName> where <ItemName is the boundary name, solution setup name, etc. For example, "BoundarySetup:PerfE1"
<PropTab>	Boundary Module: "HfssTab" Mesh Operations Module: "MeshSetupTab" Analysis Module: "HfssTab" Optimetrics Module: "OptimetricsTab" Solutions Module: *Does not support properties.* Field Overlays Module: "FieldsPostProcessorTab" Radiation Module: "RadFieldSetupTab" Circuit Module: "CCircuitTab" System Module: "SystemTab" HFSS 3D Layout Module: "HFSS 3D LayoutTab" Nexxim Module: "NexximTab" Layout elements: "BaseElementTab" Schematic elements: "ComponentTab" Optimetrics Module: "OptimetricsTab"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

For 3D Model Editor objects:

<PropServer>	Name of the object. For example, "Box1".
<PropTab>	"Geometry3DAttributeTab"

For 3D Model Editor operations:

<PropServer>	<ObjName>:<OperationName>:<int> where <int> is the operation's history index. For example, "Box2:CreateBox:2" refers to the second "CreateBox" operation in Box2's history.
<PropTab>	"Geometry3DCmdTab"

For Reporter operations on Report properties:

<PropServer>	<ReportSetup>

<ChangeProperty>	Array. For example, to set the company name in a plot header to "My Company":
	`oModule = oDesign.GetModule("ReportSetup")`
	`oModule.ChangeProperty ["NAME:AllTabs",["NAME:Header",["NAME:PropServers", "XY Plot1:Header"], ["NAME:ChangedProps",["NAME:Company Name", "Value:=", "My Company"]]]]`

Note:

For scripted property changes in the various modules and editors, refer to the chapters on the System, HFSS 3D Layout, and Nexxim tools, as well as the Layout and Schematic editors.

Callback Scripting Using PropHost Object

Callback scripts are scripts that can be set in the Property Dialog for individual properties by clicking the button in the Callback column and choosing a script that is saved with the project. Callback scripts can contain any legal script commands including general Ansys script function calls (e.g., GetApplicationName()).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

In addition, Callback scripts can also call functions on a special object named PropHost. The PropHost represents the PropServer (owner of properties) that contains the Property that is calling the Callback script. Therefore, the Callback script can use the PropHost's functions to query or set other properties in the same PropServer.

Definitions

<**propName**> = text string

<**value**> = double

<**valueText**> = text string

<**fileName**> = full path file name

<**choices**> = string containing menu choices separated by commas

<**initialChoice**> = string containing initial choice for menu; must be one of the <choices>

<**scriptName**> = string containing name of script stored in project

<**bool**> is 1 for true or 0 for false

<propType>	<propTypeName>	Property Description
0	TextProp	Text
1	MenuProp	Menu
2	CheckboxProp	Checkbox
3	VariableProp	Variable
4	VPointProp	VPoint
5	V3DPointProp	V3DPoint
6	NumberProp	Number
7	ColorProp	Color
8	PointProp	Point

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

9	ValueProp	Value
10	ButtonProp	Button
11	SeparatorProp	Separator
12	NetlistProp	Netlist
13	FileNameProp	FileName

<tabType>	**Description**	**Objects with this tab type**
0	DefaultTab	
1	PassedParameterTab	Instances of designs, components, geometric objects
2	DefinitionParameterTab	Definitions of designs, components
3	LocalVariableTab	Definitions of designs, components
4	ProjectVariableTab	Projects
5	ConstantsTab	Projects
6	BaseElementTab	Geometric objects
7	ComponentTab	Designs, components
8	PropertyTab	
9	CircuitTab	
10	SystemTab	
11	HFSS3DLayoutTab	
12	HfssTab	HFSS objects
13	OptimetricsTab	Optimetrics data
14	AltraSimTab	
15	Report3DTab	Report3d

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

16	FieldsPostProcessorTab	Fieldspostprocessor
17	MeshSetupTab	Manual meshing setup
18	RadFieldSetupTab	Radiation field geometry setup
19	Geometry3DAttributeTab	Geometry3D
20	Geometry3DCmdTab	Geometry3D
21	Geometry3DPolylineTab	Geometry3D
22	Geometry3DCSTab	Geometry3D
23	Geometry3DPlaneTab	Geometry3D
24	Geometry3DPointTab	Geometry3D
25	Geometry3DListTab	Geometry3D
26	StandardPropTab	
27	PropDisplayPropTab	
28	CustomPropTab	
39512	Component	Designs, components

The topics for this section include:

ChangeProperty for HFSS and Maxwell

ChangeProperty for Schematic and Layout

ChangeProperty (Schematic Editor and Layout Editor)

Changes properties using the ChangeProperty command. This command can be executed by oEditor to change editor properties, by oDesign to change design-level properties, and by oProject to change project-level properties. It supports creation, editing, and deletion of properties. In HFSS and Maxwell, only Variable and Separator properties can be deleted.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note: The `ChangeProperty` command supports multiple tab targets such as `DesignTab`, `ComponentTab`, and others. Property names and fields vary depending on the tab context.

Use the script recording feature to edit a property and view the resulting script entry, or use GetPropertyValue to inspect the expected format.

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><PropTabArray></td><td>Array</td><td>Structured array containing tab-level property changes. See detailed structure below.</td></tr></table>
Return Value	None

Python Syntax	ChangeProperty(["Name:AllTabs", <PropTabArray>, <PropTabArray>, ...])

Python Example	`oEditor.ChangeProperty([`
	`  "NAME:AllTabs",`
	`  ...PropTabArray entries...`
	`  ])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Property Array Structure

The following structure defines the nested arrays and parameters used in `ChangeProperty`. Each array must follow the specified order and naming convention.

```
  ["Name:<PropTab>",
    <PropServersArray>,
    <NewPropsArray>,
    <ChangedPropsArray>,
    <DeletedPropsArray>]
<PropServersArray>
  ["Name:PropServers", <PropServer>, <PropServer>, ...]
<NewPropsArray>
  ["Name:NewProps", <PropDataArray>, <PropDataArray>, ...]
<ChangedPropsArray>
  ["Name:ChangedProps", <PropDataArray>, <PropDataArray>, ...]
<DeletedPropsArray>
  ["Name:DeletedProps", <PropName>, <PropName>, ...]
<PropDataArray>
  ["NAME:<PropName>",
    "PropType:=", <PropType>,
    "NewName:=", <string>,
    "Description:=", <string>,
    "NewRowPosition:=", <int>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ReadOnly:=", <bool>,
    "Hidden:=", <bool>,
    <PropTypeSpecificArgs>]
```

`<PropType>`

Type: string

Identifies the type of property when a new property is added. In HFSS and Maxwell, only separator properties and variable properties can be added.

- `"SeparatorProp"`
- `"VariableProp"`
- `"TextProp"`
- `"NumberProp"`
- `"ValueProp"`
- `"CheckboxProp"`
- `"MenuProp"`
- `"PointProp"`
- `"VPointProp"`
- `"V3DPointProp"`
- `"ButtonProp"`

`NewName`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Specify the new name of a property if the property's name is being edited. In HFSS and Maxwell, the name can only be changed for separators and variables.

`Description`

Specify a description of the property. In HFSS and Maxwell, the description can only be changed for separators and variables.

`NewRowPosition`

Used to reorder rows in the **Property** dialog box.

In HFSS, this only applies to the **Project** > **Project Variables** panel and the **HFSS** > **DesignProperties** panel.

In Maxwell, this only applies to the **Project** > **Project Variables** panel and the **Maxwell3D** > **DesignProperties**, **Maxwell2D** > **Design Properties**, or **RMxprt** > **Design Properties** panels. Specify the new zero-based row index of the variable or separator.

`ReadOnly`

Used to mark a property as "read only" so it cannot be modified. In HFSS and Maxwell, this flag can only be set for variables and separators.

`Hidden`

Used to hide a property so it cannot be viewed outside of the **Property** dialog box. In HFSS and Maxwell, this flag can only be set for variables and separators.

```
<PropTypeSpecificArgs>

  SeparatorProp: no arguments

  TextProp: "Value:=", <string>

  NumberProp: "Value:=", <double>

  ValueProp: "Value:=", <value>

  CheckboxProp: "Value:=", <bool>

  MenuProp: "Value:=", <string>
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PointProp: "X:=", <double>, "Y:=", <double>

VPointProp: "X:=", <value>, "Y:=", <value>

V3DPointProp: "X:=", <value>, "Y:=", <value>, "Z:=", <value>

Material Button: "Material:=", <string>

Color Button: "R:=", <int>, "G:=", <int>, "B:=", <int>

Transparency Button: "Value:=", <double>

<PropTypeSpecificArgs> for VariableProps

Syntax:

"Value:=", <value>, <OptimizationFlagsArray>, <TuningFlagsArray>, <SensitivityFlagsArray>, <StatisticsFlagsArray>

Parameters:

<OptimizationFlagsArray>

["NAME:Optimization",

"Included:=", <bool>,

"Min:=", <value>,

"Max:=", <value>]

<TuningFlagsArray>

["NAME:Tuning",

"Included:=", <bool>,

"Step:=", <value>,

"Min:=", <value>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Max:=", <value>]
  <SensitivityFlagsArray>
    ["NAME:Sensitivity",
    "Included:=", <bool>,
    "Min:=", <value>,
    "Max:=", <value>,
    "IDisp:=", <value>]
  <StatisticsFlagsArray>
    ["NAME:Statistical",
    "Included:=", <bool>,
    "Dist:=", <Distribution>,
    "StdD:=", <value>,
    "Min:=", <value>,
    "Max:=", <value>,
    "Tol:=", <string>]
<Distribution>
```

Type: string

Value should be "Gaussian" or "Uniform"

`StdD`

Standard deviation.

`Min`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Low cut-off for the distribution.

`Max`

High cut-off for the distribution.

`Tol`

Tolerance for uniform distributions. Format is "<int>%". Example: "20%".

PropHost Functions

The following commands can be used to manipulate properties from a Property script.

GetEditor

GetPropHost

AddMenuProp

Use: Creates a new Menu property in tabType with name specified; choices are set to the values in choices; initial selection is initialChoice.

UI Access	NA		
Parameters	Name	Type	Description
	<tabType>	Integer	One of the following, where tab titles are shown in parentheses: 1 for PassedParameterTab ("Parameter Values") 2 for DefinitionParameterTab (Parameter Defaults") 3 for LocalVariableTab ("Variables" or "Local Variables") 4 for ProjectVariableTab ("Project variables") 5 for ConstantsTab ("Constants")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			6 for BaseElementTab ("Symbol" or "Footprint") 7 for ComponentTab ("General") 28 for CustomTab ("Intrinsic Variables") 39500 for Quantities ("Quantities") 39506 for Signals ("Signals") 39512 for Component ("Component")
	<propName>	String	Name of the new menu property
	<choices>	String	A comma-separated list of menu choice strings.
	<initialChoice>	String	One of the strings in the list of choices.
Return Value	None		

Python Syntax	AddMenuProp(<tabType>, <propName>, <choices>, <initialChoice>)
Python Example	`PropHost.AddProp(2,` `"ResChoices", "inline,upfront,parallel,series", "parallel")`

AddMenuProp2

Creates a new Menu property in tabTypeName with name specified; choices are set to the values in choices; initial selection is initialChoice.

UI Access	None		
Parameters	Name	Type	Description
	<tabType>	String	One of the following, where tab titles are shown in parentheses:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			PassedParameterTab ("Parameter Values") DefinitionParameterTab (Parameter Defaults") LocalVariableTab ("Variables" or "Local Variables") ProjectVariableTab ("Project variables") ConstantsTab ("Constants") BaseElementTab ("Symbol" or "Footprint") ComponentTab ("General") Component("Component") CustomTab ("Intrinsic Variables") Quantities ("Quantities") Signals ("Signals")
	<propName>	String	Name of the new menu property.
	<choices>	String	A comma-separated list of menu choice strings.
	<initialChoice>	String	One of the strings in the list of choices.
Return Value	None		

Python Syntax	AddMenuProp2(<tabTypeName>, <propName>, <choices>, <initialChoice>)
Python Example	`PropHost.AddMenuProp2("DefinitionParameterTab", "ResChoices", "inline,upfront,parallel,series", "parallel")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddProp

Use: Creates a new propType property in tabType with name and value specified.

<table>
<tr><td>UI Access</td><td colspan="3">NA</td></tr>
<tr><td rowspan="3">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><tabType></td><td>Integer</td><td>One of the following, where tab titles are shown in parentheses:
1 for PassedParameterTab ("Parameter Values")
2 for DefinitionParameterTab (Parameter Defaults")
3 for LocalVariableTab ("Variables" or "Local Variables")
4 for ProjectVariableTab ("Project variables")
5 for ConstantsTab ("Constants")
6 for BaseElementTab ("Symbol" or "Footprint")
7 for ComponentTab ("General")
28 for CustomTab ("Intrinsic Variables")
39500 for Quantities ("Quantities")
39506 for Signals ("Signals")
39512 for Component ("Component")</td></tr>
<tr><td><propType></td><td>Integer</td><td>One of the following to add new definition properties (e.g., to component, design or project):
0 for TextProp (any string)
1 for Menu (one,two,three)
2 for CheckboxProp ("true" or "false" values)</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			3 for VariableProp (useful for Project and Local Variables, expression) 4 for VPointProp (x and y values, which can be expressions) 5 for V3DPointProp (x, y and z values, which can be expressions) 6 for NumberProp (simple number, no expression) 7 for ColorProp (takes an RGB value) 8 for PointProp (x and y values, no expressions) 11 for SeparatorProp (used for property list organization, the value is the separator "title") 13 for FileNameProp (special button property for filenames) 39506 for GenericProp (for generics in Twin Builder, SML and VHDL) 39500 for QuantityProp (for Twin Builder quantities, SML and VHDL) 39504 for SignalProp (for VHDL signals in Twin Builder) 39502 for VariablePropNU (like a `VariableProp`, but provides access to setting NetlistUnits in Component definition dialog for Designer; for Twin Builder, use `GenericProp` in `DefinitionParametersTab`, `QuantitiyProp` in `Quantities`, and `signalProp` in `Signals`)
	<propName>	String	Name of the new menu property.
	<valueText>	String	The property value
Return Value	None		

Python Syntax	AddProp(<tabType>, <propType>, <propName>, <valueText>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`PropHost.AddProp(2, 3, "W1", "10mm")`

Note:

AddProp adds a new property using the <propName> preceded by a separator "--". For instance, a new property added with the <propName> "xxx" will be added with the name "--xxx". Consequently, all subsequent script functions that wish to access the added property must now use the name "--xxx".

For example:

PropHost.AddProp 1, 11, "Time_Domain_Options",""

Prophost.SetHidden "--Time_Domain_Options",1

propType	Property Type	Value Example
0	Text	"hello"
1	Menu	"one,two,three"
2	Checkbox	"true" or "false"
3	Variable	"34mm" or "2*val1"
4	VPoint	"x1,y1"
5	V3DPoint	"x1,y1,z1"
6	Number	"17" or "22pF"
7	Color	
8	Point	"34,27"
9	Value	"34mm" or "2*val1"
11	Separator	"colors"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

13	Filename	"c:\temp\ne22.s2p"

AddProp2

Creates a new propTypeName property in tabTypeName with name and value specified.

UI Access	NA		
Parameters	Name	Type	Description
	<tabType>	Type	One of the following, where tab titles are shown in parentheses: PassedParameterTab ("Parameter Values") DefinitionParameterTab (Parameter Defaults") LocalVariableTab ("Variables" or "Local Variables") ProjectVariableTab ("Project variables") ConstantsTab ("Constants") BaseElementTab ("Symbol" or "Footprint") ComponentTab ("General") Component("Component") CustomTab ("Intrinsic Variables") Quantities ("Quantities") Signals ("Signals")
	<propType>	String	One of the following to add new definition properties (e.g., to component, design or project):

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			TextProp (any string) CheckboxProp ("true" or "false" values) VariableProp (useful for Project and Local Variables, expression) VPointProp (x and y values, which can be expressions) V3DPointProp (x, y and z values, which can be expressions) NumberProp (simple number, no expression) ColorProp (takes an RGB value) PointProp (x and y values, no expressions) SeparatorProp (used for property list organization, the value is the separator "title") FileNameProp (special button property for filenames) GenericProp (for generics in Twin Builder, SML and VHDL) QuantityProp (for Twin Builder quantities, SML and VHDL) SignalProp (for VHDL signals in Twin Builder) VariablePropNU (like a `VariableProp`, but provides access to setting NetlistUnits in Component definition dialog for Designer; for Twin Builder, use `GenericProp` in `DefinitionParametersTab`, `QuantitiyProp` in `Quantities`, and `signalProp` in `Signals`)
	<propName>	String	Name of the new menu property.
	<valueText>	String	The property value
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddProp2(<tabTypeName>, <propTypeName>, <propName>, <valueText>)
Python Example	`PropHost.AddProp2("DefinitionParameterTab", "VariableProp", "W1", "10mm")`

ExecuteScript

Finds the named script in the Definitions/Scripts folder and runs that script; the script being run can also use the PropHost object.

UI Access	NA
Parameters	Name | Type | Description <scriptName> | String | Name of a script stored in a script library
Return Value	None

Python Syntax	ExecuteScript(<scriptName>)
Python Example	`PropHost.ExecuteScript("PropChangeScript")`

GetApplication

Returns the application object currently running the script.

UI Access	NA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description None | |
Return Value	Object The Application

Python Syntax	GetApplication()
Python Example	`oAnsoftApp2 = prophost.GetApplication()`

GetCallback

Finds named property and returns name of Callback script.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

Python Syntax	GetCallback(<propName>)
Python Example	`a = PropHost.GetCallback( "W1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChangedProperty

Use: If the script was called by a Callback associated with a property, this function returns the name of that property.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	String Returns "C" if the script was a Callback associated with the property named "C" and the script was called in response to the property "C" changing value.

Python Syntax	GetChangedProperty()
Python Example	`pn = PropHost.GetChangedProperty()`

GetDescription

Finds named property and returns description string

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDescription(<propName>)
Python Example	`a = PropHost.GetDescription( "W1")`

GetDesign

Use: Returns the interface to the specified design simulation.

Command: None

Syntax: GetDesign <DesignName>

Return Value: Interface to the specified design simulation.

Parameters: <DesignName>

Type: <string>

GetEditor

Use: Returns an interface to the editor requested IF the PropServer behind the PropHost is contained within that type of editor.

Command: None

Syntax: GetEditor(<editorName>)

Return Value: String

Python Syntax	GetEditor(<editorName>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor = GetEditor("SchematicEditor")`

GetEvaluatedText

Returns the evaluated value of the property, useful when the value is an expression.

UI Access	NA
Parameters	Name | Type | Description <propname> | String | The name of the property
Return Value	String The Evaluated Value

Python Syntax	GetEvaluatedText (<propname>)
Python Example	`ret = prophost.GetEvaluatedText("R")`

GetFileName

Finds the full path name to propName.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Full path file name or empty string.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

Directory variables can be used in the property's value (e.g. $projectdir, $userlib, $syslib, $personallib) and these will be expanded to the correct path. GetFileName always returns a path string; if propName actually contains a variable expression, that expression is evaluated to a constant string before returning.

Python Syntax	GetFileName(<propName>)
Python Example	`a = PropHost.GetFileName("SubstrateFile")`

GetHidden

Finds named property and returns its Hidden flag.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Returns 1 if property is hidden and 0 if it is not.

Python Syntax	GetHidden(<propName>)
Python Example	`a= PropHost.GetHidden( "W1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProgress

Use: Returns the percentage (from 0 to 100) of the simulation completed.

Command: None

Syntax: GetProgress(<simProgress>)

Return Value: String

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

GetPropServers

Returns array of objects that have properties showing on tabTypeName.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Array of propserver names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetPropServers(<tabTypeName>)
Python Example	`objects = PropHost.GetPropServers("PassedParameterTab")` Returns array containing PropServers that have properties shown on PassedParameterTab; this would include only components and designs; individual properties can be accessed using standard notation, e.g. objects(0) might contain "CompInst@CAP_;1".

GetPropTabType

Finds named property and returns the id of the tab it is in.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
Return Value	Returns string		

Python Syntax	GetPropTabType(<propName>)
Python Example	`a = PropHost.GetPropTabType( "W1")`

GetReadOnly

Finds named property and returns its ReadOnly flag.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr></table>
Return Value	Returns 1 if property is read-only and 0 if it is not.

Python Syntax	GetReadOnly(<propName>)
Python Example	`a = PropHost.GetReadOnly( "W1");` `returns 1`

GetRunStatus

Use: Returns the status number of the specified design simulation.

Command: None

Syntax: GetRunStatus(<statusNumber>)

Return Value: Returns string.

GetTabTypeName

Finds named property and returns the name of the tab it is on.

UI Access	NA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

Python Syntax	GetTabTypeName(<propName>)
Python Example	`a = PropHost.GetTabTypeName( "W1")`

GetText

Finds property in any tab and returns its value as a text string.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

Python Syntax	GetText(<propName>)
Python Example	`a = PropHost.GetText("C")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetValue

Use: Finds property in any tab and returns its value as a double.

Command: None

Syntax: GetValue(<propName>)

Return Value: Returns double.

> **Note:**
>
> Values are returned in SI units. Compound SI units are, in general, not supported. Temperature values are returned in Celsius

IsValueConstant

Determines whether the specified property is a constant or not. Returns True if propName is evaluated to be a constant; returns False if propName is evaluated to be an expression.

UI Access	NA		
Parameters	Name	Type	Description
	<propname>	String	The name of the property.
Return Value	Integer; non-zero for constant, zero for variable or expression.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	IsValueConstant (<propname>)
Python Example	`ret = prophost.IsValueConstant("R")`

PropertyExists

Finds named property and returns its property type.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Returns 1 if property exists in any tab, 0 if it does not.

Python Syntax	PropertyExists(<propName>)
Python Example	`a = PropHost.PropertyExists( "W1")`

RemoveProp

Removes the named property from whichever tab it is found.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	RemoveProp(<propName>)
Python Example	`PropHost.RemoveProp("W1")`

SetCallback

Finds named property and sets its Callback script.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<scriptName>	String	Name of a script stored in a script library.
Return Value	None		

Python Syntax	SetCallback(<propName>, <scriptName>)
Python Example	`PropHost.SetCallback("W1", "SynchronizeResistors")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetDescription

Finds named property and sets its description text.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<description>	String	Descriptive text.
Return Value	None		

Python Syntax	SetDescription(<propName>, <description>)
Python Example	`PropHost.SetDescription("W1", "this is the width of the gate")`

SetHidden

Finds named property and sets its Hidden flag.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<hidden>	Boolean	True or 1, false or 0.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SetHidden(<propName>, <hidden>)
Python Example	`PropHost.SetHidden("W1", 1)`

SetReadOnly

Finds named property and sets its ReadOnly flag.

UI Access	NA
Parameters	Name / Type / Description <propName> / String / Name of the property <hidden> / Boolean / True or 1, false or 0.
Return Value	None

Python Syntax	SetReadOnly(<propName>, <hidden>)
Python Example	`PropHost.SetReadOnly("W1", 1)`

SetText

Finds property in any tab and sets its value to a text string.

UI Access	NA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr><tr><td><text></td><td>String</td><td>The value for the property.</td></tr></table>
Return Value	None

Python Syntax	SetText(<propName>, <text>)
Python Example	`PropHost.SetText("C","22nF")`

SetValue

Finds property in any tab and sets its value to a double.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr><tr><td><value></td><td>Real</td><td>A real number for the property value.</td></tr></table>
Return Value	None

Python Syntax	SetValue(<propName>, <value>)
Python Example	`PropHost.SetValue("C",2e-9)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeProperty (HFSS and Maxwell)

Changes to properties are scripted using the ChangeProperty command. This command can be executed by the `oEditor` to change editor properties, by the `oDesign` to change design level properties, and by the `oProject` to change project-level properties. The command can be used to create, edit, and/or remove properties. In HFSS and Maxwell, only `Variable` and `Separator` properties can be deleted.

Use the script recording feature and edit a property, and then view the resulting script entry or use GetPropertyValue for the desired property to see the expected format.

<table>
<tr><td>UI Access</td><td>Not available</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Name></td><td>String</td><td>"Name:AllTabs"</td></tr>
<tr><td><PropTabArray></td><td>Array</td><td>["Name:<PropTab>",
<PropServersArray>,
<NewPropsArray>,
<ChangedPropsArray>,
<DeletedPropsArray>]
Multiple <PropTabArray> can be added</td></tr>
<tr><td><PropServersArray></td><td>String</td><td>["Name:PropServers", <>, <PropServer>, ...]
Note that PropServer is an object identifier, generally returned from another script method, such as CompInst@R;2;3</td></tr>
<tr><td><NewPropsArray></td><td>Array</td><td>["Name:NewProps", <PropDataArray>, <PropDataArray>,...]
See below for more details on <PropDataArray>.</td></tr>
<tr><td><ChangedPropsAr-</td><td>Array</td><td>["Name:ChangedProps", <PropDataArray>, <PropDataAr-</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	ray>		ray>, ...]
	<DeletedPropsArray>	Array	*["Name:DeletedProps", <PropName>,*<PropName>, ...)
Return Value	None		

Python Syntax	ChangeProperty ["Name:AllTabs", <*PropTabArray*>, <*PropTabArray*>, ...] ChangeProperty(<modulename>:<setup name>:<sweep name>)

Python Example	2D Maxwell Example of changing solve setup properties:
	`oDesign.ChangeProperty(`
	`  [`
	`  "NAME:AllTabs",`
	`  [`
	`  "NAME:Solver",`
	`  [`
	`  "NAME:PropServers",`
	`  "AnalysisSetup:Setup1"`
	`  ],`
	`  [`
	`  "NAME:ChangedProps",`
	`  [`
	`  "NAME:Scalar Potential",`
	`  "Value:=" , "Second Order"`
	`  ]`
	`  ]`
	`  ]`
	`  ])`

<PropDataArray>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:<PropName>",
   "PropType:=", <PropType>,
   "NewName:=", <string>,
   "Description:=", <string>,
   "NewRowPosition:=", <int>,
   "ReadOnly:=", <bool>,
   "Hidden:=", <bool>,
   <PropTypeSpecificArgs>
]
```

<PropType>

String that Identifies the type of property when a new property is added. In HFSS and Maxwell, only separator properties and variable properties can be added. Possible values are

"SeparatorProp"

"VariableProp"

"TextProp"

"NumberProp"

"ValueProp"

"CheckboxProp"

"MenuProp"

"PointProp"

"VPointProp"

"V3DPointProp"

"ButtonProp"

<NewName>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Specify the new name of a property if the property's name is being edited. In HFSS and Maxwell, the name can only be changed for separators and variables.

<Description>

Specify a description of the property. In HFSS and Maxwell, the description can only be changed for separators and variables.

<NewRowPosition>

Used to reorder rows in the **Property** dialog box; specify the new zero-based row index of the variable or separator.

In HFSS, this only applies to the **Project** > **Project Variables** panel and the **HFSS** > **DesignProperties** panel.

In Maxwell, this only applies to the **Project** > **Project Variables** panel and the **Maxwell3D** > **DesignProperties**, **Maxwell2D** > **Design Properties**, or **RMxprt** > **Design Properties** panels.

<ReadOnly>

Used to mark a property as "read only" so it can not be modified. In HFSS and Maxwell, this flag can only be set for variables and separators.

<Hidden>

Used to hide a property so it can not be viewed outside of the **Property** dialog box. In HFSS and Maxwell, this flag can only be set for variables and separators.

<PropTypeSpecificArgs>

```
SeparatorProp: no arguments
TextProp: "Value:=", <string>
NumberProp: "Value:=", <double>
ValueProp: "Value:=", <value>
CheckboxProp: "Value:=", <bool>
MenuProp: "Value:=", <string>
PointProp "X:=", <double>, "Y:=", <double>
VPointProp: "X:=", <value>, "Y:=", <value>
V3DPointProp: "X:=",<value>, "Y:=",<value>, "Z:=",<value>
Material Button: "Material:=", <string>
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Color Button: "R:=",<int>,"G:=",<int>,"B:=",<int>
Transparency Button: "Value:=", <double>
```

<PropTypeSpecificArgs> for VariableProps

Syntax:

```
"Value:=", <value>, <OptimizationFlagsArray>,
<TuningFlagsArray>, <SensitivityFlagsArray>,
<StatisticsFlagsArray>
```

Parameters:

```
<OptimizationFlagsArray>
  ["NAME:Optimization",
  "Included:=", <bool>,
  "Min:=", <value>,
  "Max:=", <value>]


<TuningFlagsArray>
  ["NAME:Tuning",
  "Included:=", <bool>,
  "Step:=", <value>,
  "Min:=", <value>,
  "Max:=", <value>]


<SensitivityFlagsArray>
  ["NAME:Sensitivity",
  "Included:=", <bool>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Min:=", <value>,
    "Max:=", <value>,
    "IDisp:=", <value> ]


  <StatisticsFlagsArray>

    ["NAME:Statistical",
    "Included:=", <bool>,
    "Dist:=", <Distribution>, # String, value should be "Gaussian" or "Uniform"
    "StdD:=", <value>, # Standard deviation
    "Min:=", <value>, # Low cut-off for the distribution
    "Max:=", <value>, # High cut-off for the distribution
    "Tol:=", <string>] # Tolerance for uniform distributions. Format is "<int>%"; for example:
    "20%"
```

GetArrayVariables

Returns a list of array variables. To get a list of indexed project variables, execute with oProject. To get a list of indexed local variables, use oDesign.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing names of variables.

Python Syntax	GetArrayVariables()
Python Example	`oProject.GetArrayVariables()` `oDesign.GetArrayVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProperties

Gets a list of all the properties belonging to a specific <PropServer> and <PropTab>. This can be executed by the oProject, oDesign, or oEditor variables.

<table>
<tr><td>UI Access</td><td colspan="2">N/A</td></tr>
<tr><td>Parameters</td><td colspan="2"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PropTab></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:
PassedParameterTab ("Parameter Values")
DefinitionParameterTab (Parameter Defaults")
LocalVariableTab ("Variables" or "Local Variables")
ProjectVariableTab ("Project variables")
ConstantsTab ("Constants")
BaseElementTab ("Symbol" or "Footprint")
ComponentTab ("General")
Component("Component")
CustomTab ("Intrinsic Variables")
Quantities ("Quantities")
Signals ("Signals")</td></tr>
<tr><td><PropServer></td><td>String</td><td>An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="2">Array of strings containing the names of the appropriate properties.</td></tr>
</table>

Python Syntax	GetProperties(<*PropTab*>, <*PropServer*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.GetProperties('PassedParameterTab', 'k')`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

> **Tip:** Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PropTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<*PropServer*>	String	An object identifier, generally returned from another script method, such as `Comp‡nst@R;2;3`
	<*PropName*>	String	Name of the property.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	String value of the property.

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`
	`...`

GetVariables

Returns a list of all defined variables. To get a list of project variables, execute this command using `oProject`. To get a list of local variables, use `oDesign`.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the variables.

Python Syntax	GetVariables ()
Python Example	`oProject.GetVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oDesign.GetVariables()`

GetVariableValue

Gets the value of a single specified variable. To get the value of project variables, execute this command using `oProject`. To get the value of local variables, use `oDesign`.

UI Access	N/A
Parameters	Name | Type | Description <*VarName*> | String | Name of the variable to access.
Return Value	String represents the value of the variable.

Python Syntax	GetVariableValue(<*VarName*>)
Python Example	`oProject.GetVariableValue("var_name")`

SetPropertyValue

Sets the value of a single property belonging to a specific PropServer and PropTab. This function is available with the Project, Design or Editor objects, including definition editors.This is not supported for properties of the following types: ButtonProp, PointProp, V3DPointProp, and VPointProp. Only the ChangeProperty command can be used to modify these properties.

Use the script recording feature and edit a property, and then view the resulting script entry or use GetPropertyValue for the desired property to see the expected format.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*propTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
<*propServer*>	String	An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3
<*propName*>	String	Name of the property.
<*propValue*>	String	The value for the property

Return Value: None.

Python Syntax	SetPropertyValue(<*propTab*>, <*propServer*>, <*propName*>, <*propValue*>)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetVariableValue

Sets the value of a variable. To set the value of a project variable, execute this command using `oProject`. To set the value of a local variable, use `oDesign`.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*VarName*></td><td>String</td><td>Variable name.</td></tr><tr><td><*VarValue*></td><td>Value</td><td>New value for the variable.</td></tr></table>
Return Value	None.

Python Syntax	SetVariableValue (<*VarName*>, <*VarValue*>)
Python Example	`oProject.SetVariableValue('$Var1', '3mm')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

3 - Application Object Script Commands

The Application object commands permit you to get the AppDesktop. Application object commands should be executed by the oAnsoftApp object.

```
oAnsoftApp.<CommandName> <args>
```

General Application Script Commands

The following are general script commands recognized by the **oAnsoftApp** object:

- GetAppDesktop

The following deprecated commands are no longer supported and produce an error if used.

- GetDesiredRamMBLimit (deprecated)
- GetHPCLicenseType (deprecated)
- GetMaximumRamMBLimit (deprecated)
- GetMPISpawnCmd(deprecated)
- GetMPIVendor (deprecated)
- GetNumberOfProcessors (deprecated)
- GetUseHPCForMP (deprecated)
- SetDesiredRamMBLimit (deprecated)
- SetHPCLicenseType (deprecated)
- SetMaximumRamMBLimit (deprecated)
- SetMPISpawnCmd (deprecated)
- SetMPIVendor (deprecated)
- SetNumberOfProcessors (deprecated)
- SetUseHPCForMP (deprecated)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAppDesktop

GetAppDesktop is a function of oAnsoftApp. This function does not take an input and it returns an object. The object is assigned to the variable oDesktop.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Object

Python Syntax	GetAppDesktop()
Python Example	`oDesktop = oAnsoftApp.GetAppDesktop()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

4 - Desktop Object Script Commands

Desktop commands should be executed by the oDesktop object. Some new commands permit you to query objects when you do not know the names. See: Object Oriented Property Scripting.

```
oDesktop = CreateObject("Ansoft.ElectronicsDesktop")

oDesktop.CommandName <args>
```

AddMessage

AreThereSimulationsRunning

ClearMessages

CloseAllWindows

CloseProject

CloseProjectNoForce

DeleteProject

DeleteRegistryEntry

DoesRegistryValueExist

DownloadJobResults

EnableAutoSave

ExportOptionsFiles

GetActiveProject

GetActiveScheduler

GetActiveSchedulerInfo

GetAutoSaveEnabled

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetBuildDateTimeString

GetCustomMenuSet

GetDefaultUnit

GetDesktopConfiguration

GetDistributedAnalysisMachines

GetDistributedAnalysisMachinesForDesignType

GetExeDir

GetGDIObjectCount

GetLibraryDirectory

GetLocalizationHelper

GetMessages

GetMonitorData

GetPersonalLibDirectory

GetProcessID

GetProjectDirectory

GetProjectList

GetProjects

GetRegistryInt

GetRegistryString

GetRunningInstancesMgr

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSchematicEnvironment

GetScriptingToolsHelper

GetSysLibDirectory

GetTempDirectory

GetUserLibDirectory

GetVersion

IsFeatureEnabled

LaunchJobMonitor

NewProject

OpenAndConvertProject

OpenProject

OpenProjectWithConversion

PageSetup

PauseRecording

PauseScript

Print

QuitApplication

RefreshJobMonitor

ResetLogging

RestoreProjectArchive

RestoreWindow

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ResumeRecording

RunACTWizardScript

RunProgram

RunScript

RunScriptWithArguments

SelectScheduler

SetActiveProject

SetActiveProjectByPath

SetCustomMenuSet

SetDesktopConfiguration

SetLibraryDirectory

SetProjectDirectory

SetRegistryFromFile

SetRegistryInt

SetRegistryString

SetSchematicEnvironment

SetTempDirectory

ShowDockingWindow

Sleep

StopSimulations

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SubmitJob

TileWindows

Related Topics:

Desktop Commands For Registry Values

ImportExport Tool Commands

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddMessage

Add a message with severity and context to message window.

UI Access	N/A
Parameters	

Name	Type	Description
<*projectName*>	String	Project name. Passing an empty string adds the message as the desktop global message.
<*designName*>	String	Design name. Ignored if project name is empty. Passing an empty string adds the message to project node in the message tree.
<*severity*>	Integer	One of "Error", "Warning" or "Info". Anything other than the first two is treated as "Info" 0 = Informational, 1 = Warning, 2 = Error, 3 = Fatal
<*msg*>	String	The message for the message window.
<*category*>	String	Optional. The category is created with the message under the design tree node if the category does not exist. If the category already exists, the new message is added to the end of the existing category. It is ignored if the project or design is empty. If missing or empty, the message is added to the Design node in the message tree.

Return Value	None.

Python Syntax	AddMessage(<*projectName*>, <*designName*>, <*severity*>, <*msg*>, <*category*>)
Python Example	`oDesktop.AddMessage("Project1", "HFSS 3D LayoutDesign", 0, "This is a test message", "")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AreThereSimulationsRunning

Returns a bool specifying whether there are simulations running, or either running or pending, depending on the inclusion of alsoPending.

UI Access	NA
Parameters	Name: <bool> <AlsoPending> — Type: Integer — Description: Whether to report if simulations are running, or running and pending.
Return Value	A human readable status string specifying what happened.

Python Syntax	AreThereSimulationsRunning (<bool>, clean)
Python Example	`oDesktop.AreThereSimulationsRunning(bool clean)`

ClearMessages

For a specified project and design, this command clears all messages in a specified severity range. The user-specified integral severity level is interpreted as:

```
0 => info, 1 => warning, 2 => error, and 3 => fatal error.
```

The ClearMessages function accepts four input arguments. The first two specifies project and design, respectively. This function clears messages in the range specified by the third (interpreted as start severity) and fourth (interpreted as stop severity) arguments. The fourth argument has a default value of 0. The last two arguments need not be specified in any given order, i.e., they can indicate either ascending or descendingtrend. The function clears all messages having severity level within this specified range. The start and stop severity need not be specified in any given order (i.e., they can indicate either ascending or descending severity).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>In Message Manager, right-click and select Clear messages for [ProjectName]...</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><projectName></td><td>String</td><td>Name of the project from which to clear messages.</td></tr>
<tr><td><designName></td><td>String</td><td>Name of the design under the <projectName> from which to clear messages.</td></tr>
<tr><td><startSeverity></td><td>Integer</td><td>User-specified severity level at which messages start getting cleared:
0 = informational
1 = warning
2 = error
3 = fatal error</td></tr>
<tr><td><stopSeverity></td><td>Integer</td><td>Optional. User-specified stop severity level until which messages will be cleared.
0 = informational
1 = warning
2 = error
3 = fatal error
If not specified, <stopSeverity> has a default value of 0.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>ClearMessages(<<i>projectName</i>>, <<i>designName</i>>, <<i>startSeverity</i>>, <<i>stopSeverity</i>>)</td></tr>
<tr><td>Python Example</td><td>Preserve Current Functionality; stopSeverity = 0 (DEFAULT VALUE)
<pre># startSeverity = 0; clears just the Info messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0)</pre></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# startSeverity = 1; clears all the Info and Warning messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 1)
# startSeverity = 2; clears all the Info, Warning, Error
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 2)
# startSeverity = 3; clears all messages, i.e., Info, Warning, Error, Fatal
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 3)
```

Extend the current functionality by enabling range-based deletion

```
# startSeverity = 0; clears the Info messages;
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0)
# startSeverity = 0 and stopSeverity = 0; clears all the Info messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0,
0)
# startSeverity = 0 and stopSeverity = 1; clears all the Info and Warning mes-
sages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0,
1)
# startSeverity = 0 and stopSeverity = 2; clears all the Info, Warning, and Error
messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0,
2)
# startSeverity = 0 and stopSeverity = 3; clears all the Info, Warning, Error,
and Fatal messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0,
3)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# startSeverity = 1; clears all the Info and Warning messages;
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",1)
# startSeverity = 1 and stopSeverity = 1; clears all the Warning messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",1, 1)
# startSeverity = 1 and stopSeverity = 2; clears all the Warning and Error mes-
sages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",1, 2)
# startSeverity = 1 and stopSeverity = 3; clears all the Warning, Error, and
Fatal messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",1, 3)


# startSeverity = 2; clears all the Info, Warning, and Error messages;
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",2)
# startSeverity = 2 and stopSeverity = 2; clears all the Error messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",2, 2)
# startSeverity = 2 and stopSeverity = 3; clears all the Error and Fatal messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",2, 3)
# startSeverity = 2 and stopSeverity = 0; clears all the Info, Warning, and Error
messages
oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",2, 0)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

	``
	# startSeverity = 3; clears all the Info, Warning, Error, and Fatal messages;
	oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",3)
	# startSeverity = 3 and stopSeverity = 3; clears all the Fatal error messages
	oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",3, 3)
	# startSeverity = 3 and stopSeverity = 0; clears all the Info, Warning, Error, and Fatal messages
	oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor","Maxwell3DDesign1",3, 0)
	``

CloseAllWindows

Closes all MDI child windows on the desktop.

UI Access	From main menu, **Window** > **CloseAll**.
Parameters	None.
Return Value	None.

Python Syntax	CloseAllWindows()
Python Example	`oDesktop.CloseAllWindows()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CloseProject

Closes a specified project. Changes to the project are not saved. Save the project using the Project command **Save** or **Save As** before closing to save changes.

UI Access	**File** > **Close**
Parameters	Name | Type | Description <ProjectName> | String | The name of the project already in the Desktop that is to be closed, without path or extension
Return Value	None

Python Syntax	CloseProject (<*ProjectName*>)
Python Example	`oDesktop.CloseProject("MyProject")`

CloseProjectNoForce

Closes a named project currently open in the Desktop, unless a simulation is running. Changes to the project will not be saved. Save the project using the Project command **Save** or **Save As** before closing to save changes. To determine if the project has been closed, use `GetProjectList` and see if the named project is present.

UI Access	**File** > **Close**
Parameters	Name | Type | Description <ProjectName> | String | The name of the project already on the Desktop that is to be closed, without

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			path or extension
Return Value	None		

Python Syntax	CloseProjectNoForce (<*ProjectName*>)
Python Example	`oDesktop.CloseProjectNoForce("MyProject")`

Count

Gets the total number of queried projects or designs obtained by GetProjects() and GetDesigns() commands.

UI Access	N/A
Parameters	None
Return Value	Integer

Python Syntax	Count

Python Example	Example with three projects open in the Electronic Desktop:
	`projects = oDesktop.GetProjects()`
	`numprojects = projects.Count`
	`numprojects`
	`3`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteProject

Deletes a project from disk.

UI Access	**Edit > Delete**
Parameters	Name | Type | Description <ProjectName> | String | Name of the project
Return Value	None

Python Syntax	DeleteProject (<*ProjectName*>)
Python Example	`oDesktop.DeleteProject("MyProject")`

DownloadJobResults

This command is for downloading results from Ansys Cloud. Before using this script command, the command SelectScheduler() must be used first to select “ansys cloud” scheduler. This makes sure that current scheduler is Ansys Cloud, and user is logged in. Then, either a valid .q file or .q.completed file must be in the project folder, or, a valid job ID and a “batchinfo” folder containing the corresponding .jobid file are required in the project folder. When the download requirements are met, the command downloads results from Ansys Cloud using the specified filters to the given folder.

UI Access	Select Scheduler
Parameters	Name | Type | Description <jobID> | String | Provide the job ID of the target job. The job ID must be able to be found in cur-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			rent .q (or .q.completed) file, or inside the “batchinfo” folder in projectPath. If the job ID is empty, the job ID in current .q (or .q.completed) file will be used.
	<projectPath>	String	A string of path to locate the project folder. The project file may be not necessary, but .q (or .q.completed) file or “batchinfo” folder with valid .jobid files are required.
	<resultPath>	String	A string giving the folder path for the download to save to.
	<Filters> (optional)	String	A string containing filters to download. The delimiter of file types is “;”. If no filter specified, the default filter “*” will be applied, which requests all files for download.
Return Value	A Boolean result about download complete or not		

Python Syntax	DownloadJobResults(*jobID*, *projectPath*, *resultsPath*, *filters*)
Python Example	`boolDownloadCompleted = oDesktop.DownloadJobResults("01234567890123456789", "C:\\projects\\basic.aedt", "C:\\projects\\DownloadResults\\", "*")`

DeleteRegistryEntry

Deletes a registry entry from a registry key. Returns true if deletion succeeded.

UI Access	N/A		
Parameters	Name	Type	Description
	<pathToRegistry>	String	Path to Registry entry.
Return Value	Boolean: • **True** – Key has been deleted. • **False** – Key does not exist.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteRegistryEntry(<*pathToRegistry*>)
Python Example	`res = oDesktop.DeleteRegistryEntry("Desktop/ColorScheme")`

DoesRegistryValueExist

Determines whether a registry value exists.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*KeyName*></td><td>String</td><td>Full name of registry key, including path.</td></tr></table>
Return Value	Boolean: • **True** – Key exists. • **False** – Key does not exist.

Python Syntax	DoesRegistryValueExist(<*KeyName*>)
Python Example	`Exist = oDesktop.DoesRegistryValueExist('Desktop/ActiveDSOConfigurations/HFSS 3D Layout')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnableAutoSave

Enable or disable autosave feature.

UI Access	N/A
Parameters	Name | Type | Description <*enable*> | Boolean | True to enable autosave; False disables it.
Return Value	None

Python Syntax	EnableAutoSave(<*enable*>)
Python Example	`oDesktop.EnableAutoSave(True)`

ExportOptionsFiles

Copies the options config files to the *DestinationDirectory*.

UI Access	**Tools** > **Options** > **Export Options Files ...**
Parameters	Name | Type | Description <DestinationDirectory> | String | The path to the destination directory.
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportOptionsFiles(*<DestinationDirectory>)*
Python Example	`oDesktop.ExportOptionsFiles("D:/test/export/")`

GetActiveProject

Obtains the project currently active in the Desktop, as an object.

> **Note:**
>
> **GetActiveProject** returns normally if there are no active objects.

UI Access	N/A
Parameters	None.
Return Value	Object: the project that is currently active in the desktop.

Python Syntax	GetActiveProject()
Python Example	`oProject = oDesktop.GetActiveProject()`

GetActiveScheduler

Obtains the name of the scheduler active in the Desktop.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

GetActiveScheduler returns normally if there are no active objects.

UI Access	N/A
Parameters	None.
Return Value	Name of the scheduler that is currently active in the desktop. Gets results such as "RSM", "Remote RSM", "Ansys Cloud", "SLURM", "LSF", "PBS", "SGE", "Windows HPC", etc. (User-friendly scheduler names.)

Python Syntax	GetActiveScheduler()
Python Example	`oSchedulerName = oDesktop.GetActiveScheduler()`

GetActiveSchedulerInfo

Obtains info for the scheduler active in the Desktop.

Note:

GetActiveSchedulerInfo returns normally if there are no active objects.

UI Access	N/A
Parameters	None.
Return Value	Info for the scheduler that is currently active in the desktop or a remote scheduler. Gets results such as "RSM",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Remote RSM", "Ansys Cloud", "SLURM", "LSF", "PBS", "SGE", "Windows HPC", as well as server info used to select a server. For a local scheduler, the return format resembles '{"Selected Scheduler":"RSM","Scheduler Name":"RSM","Server":""}'. For a remote scheduler, the return format resembles '{"Selected Scheduler":"Remote RSM","Scheduler Name":"SLURM","Server":"<server>"}'

Python Syntax	GetActiveSchedulerInfo()
Python Example	`oSchedulerInfo = oDesktop.GetActiveSchedulerInfo()`

GetAutoSaveEnabled

Checks whether the autosave feature is enabled.

UI Access	N/A
Parameters	None.
Return Value	Integer: • 1 – Autosave is enabled. • 0 – Autosave is not enabled.

Python Syntax	GetAutoSaveEnabled()
Python Example	`Enabled = oDesktop.GetAutoSaveEnabled()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetBuildDateTimeString

Returns a string representing the build date and time of the product.

UI Access	N/A
Parameters	None.
Return Value	String build date and time, in the format: year-month-day hour:minute:second. Example: 2025-01-18 21:59:33

Python Syntax	GetBuildDateTimeString()
Python Example	`oDesktop.GetBuildDateTimeString()`

GetCustomMenuSet

Returns the name of the current selected menu set in **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**.

UI Access	N/A
Parameters	None.
Return Value	String containing current menu set. For example, 'Default', 'EM', ‘Twin Builder’.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetCustomMenuSet ()
Python Example	`oDesktop.GetCustomMenuSet()`

GetDefaultUnit

Returns the default unit for a physical quantity.

UI Access	**Tools** > **Options** > **General Options** > **Default Units**. Note that this menu only displays units that can be changed, while the script can be used to view additional default units.

Parameters

Name	Type	Description
<type>	String	String containing a type of measurement. Valid strings are (case insensitive): • "Acceleration" • "Angle" • "AngularAcceleration" • "AngularDamping" • "AngularSpeed" • "Capacitance" • "Conductance" • "Current" • "CurrentChangeRate" • "DataRate" • "DeltaH" (Magnetic Field Strength)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• "Density" • "Flux" • "Force" • "Frequency" • "Inductance" • "Length" • "MagneticReluctance" • "Mass" • "MassFlowRate" • "MomentInertia" • "Power" • "Pressure" • "PressureCoefficient" • "Resistance" • "SaturateMagnetization" (Magnetic Inductance) • "Speed" • "Temperature" • "Time" • "Torque" • "Voltage" • "VoltageChangeRate" • "Volume" • "VolumeFlowPerPressureRoot" • "VolumeFlowRate"
Return Value	String containing the default unit (for example, "mm").		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDefaultUnit(<type>)
Python Example	`oDesktop.GetDefaultUnit("Length")`

GetDesktopConfiguration

Returns the name of the current selected configuration in **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**.

UI Access	N/A
Parameters	None.
Return Value	String containing current Desktop configuration. For example, 'All', ‘Twin Builder’.

Python Syntax	GetDesktopConfiguration()
Python Example	`oDesktop.GetDesktopConfiguration()`

GetDistributedAnalysisMachines

Gets a list of machines used for distributed analysis. You can iterate through the list using standard scripting methods.

UI Access	N/A
Parameters	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Returns a collection of names of machines used for distributed analysis.

Python Syntax	GetDistributedAnalysisMachines()
Python Example	`oDesktop.GetDistributedAnalysisMachines()`

GetDistributedAnalysisMachinesForDesignType

To obtain a list of the machines set up for analysis of the specified design type.

UI Access	NA
Parameters	Name: <designTypeName> — Type: String — Description: The name of the type of design, such as "Twin Builder", "HFSS", HFSS-IE", "Maxwell 3D" , "Maxwell 2D", "RMxprt", "EM Design", "Circuit", "System", "Q3D Extractor", "2D Extractor"
Return Value	Object; returns a collection of machine names.

Python Syntax	GetDistributedAnalysisMachinesForDesignType (<designTypeName>)
Python Example	`machineNames =oDesktop.GetDistributedAnalysisMachinesForDesignType ("HFSS 3D Layout")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetExeDir

Returns the path where the executable is located.

UI Access	N/A
Parameters	None.
Return Value	String path where executable is located. Example: "C:\Program Files\ANSYS Inc\v261\AnsysEM"

Python Syntax	GetExeDir()
Python Example	`oDesktop.GetExeDir()`

GetGDIObjectCount

> **Note:**
> This command is for internal Ansys use only.

Python Syntax	GetGDIObjectCount()
Python Example	`oDesktop.GetGDIObjectCount()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetLibraryDirectory

Get the path to the `SysLib` directory.

UI Access	NA
Parameters	Name: None; Type: ; Description:
Return Value	String The path to the SysLib directory.

Python Syntax	GetLibraryDirectory()
Python Example	`AddInfoMessage(str(oDesktop.GetLibraryDirectory()))`

GetLocalizationHelper

> **Note:**
> This command is for internal Ansys use only.

Returns the object for the localization helper.

UI Access	NA
Parameters	Name: None; Type: ; Description:
Return Value	Object

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Localization helper object, such as "IDispatch(ILocalizationHelper)"

Python Syntax	GetLocalizationHelper()
Python Example	`oDesktop.GetLocalizationHelper()`

GetMessages

Get the messages from a specified project and design.

UI Access	NA
Parameters	See table below
Return Value	Array of string messages.

Name	Type	Description
<*ProjectName*>	String	Name of the project for which to collect messages. An incorrect project name results in no messages (design is ignored). An empty project name results in all messages (design is ignored)
<*DesignName*>	String	Name of the design in the named project for which to collect messages. An incorrect design name results in no messages for the named project. An empty design name results in all messages for the named project
<*Severity*>	Integer	Severity is 0-3, and is tied in to info/warning/error/fatal types as follows: • 0 – info and above • 1 – warning and above • 2 – error and fatal • 3 – fatal only (rarely used)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetMessages (<*ProjectName*>, <*DesignName*>, <*Severity*>)
Python Example	`Messages = oDesktop.GetMessages("MyProject", "HFSS 3D Layout1", 1)`

GetMonitorData

Get monitor data. This command is requires a -monitor flag on the ansysedt.exe command line and is used only with the web client.

UI Access	NA
Parameters	

Name	Type	Description
<Request>	String	a json string describing what monitor data is desired. This is a list of triples separated by spaces of the form: `type startposition maxnumber`. The `type` must be one of convergence profile `sweptvar` progress variations, `displaytype`, or a guide that was previously returned in a displaytype block. The `startposition` is an integer that specifies where to start in the sequence of monitor data of that type. Each of these types is available in a list, ordered by time. For example, as the simulation goes on, data points are added to the internal list of convergence items. If the caller needs to incrementally update a display, it can repeatedly call this method with the `startposition` for each of the types given as the size of the list it has previously obtained and displayed, which ensures there will be no duplication and the returned data will be new each time. The `maxnumber`, if greater than 0, is the limit on the number of new items returned for each type. If the caller is able to handle any number of returned values, it can be left at 0.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	a json string with the results.

Python Syntax	GetMonitorData (<Request>)
Python Example	`Messages = oDesktop.GetMonitorData(request)`

GetPersonalLibDirectory

Get the path to the `PersonalLib` directory.

UI Access	N/A
Parameters	None.
Return Value	String path to the `PersonalLib` directory.

Python Syntax	GetPersonalLibDirectory()
Python Example	`oDesktop.GetPersonalLibDirectory()`

GetProcessID

Returns the process ID of ansysedt.exe.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	Integer process ID of ansysedt.exe. For example, 12716.

Python Syntax	GetProcessID ()
Python Example	`oDesktop.GetProcessID()`

GetProjectDirectory

Gets the path to the Project directory.

UI Access	N/A
Parameters	None.
Return Value	String path to the Project directory.

Python Syntax	GetProjectDirectory()
Python Example	`oDesktop.GetProjectDirectory()`

GetProjectList

Returns a list of all projects that are open in Electronics Desktop.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the names of all open projects in Electronics Desktop.

Python Syntax	GetProjectList()
Python Example	`list_of_projects = oDesktop.GetProjectList()`

GetProjects

Returns a list of all the projects that are currently open in Electronics Desktop. Once you have the projects, you can iterate through them using standard scripting methods.

UI Access	N/A
Parameters	None.
Return Value	Returns a collection containing objects for all open projects in Electronics Desktop.

Python Syntax	GetProjects()
Python Example	`oDesktop.GetProjects()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetRegistryInt

Obtains registry key integer value.

UI Access	N/A
Parameters	Name: *<KeyName>* — Type: String — Description: Full name of registry key, including path.
Return Value	Integer if the integer value is found. Return as Bad-Argument-Value if registry key does not exist or it is not an integer value.

Python Syntax	GetRegistryInt(*<KeyName>*)
Python Example	`num = oDesktop.GetRegistryInt('Desktop/Settings/ProjectOptions/HFSS 3D Layout/UpdateReportsDynamicallyOnEdits')`

GetRegistryString

Obtains registry key string value.

UI Access	N/A
Parameters	Name: *<KeyName>* — Type: String — Description: Full name of registry key, including path.
Return Value	String if the string value is found. Return as Bad-Argument-Value if registry key does not exist or it is not a string value.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetRegistryString(<*KeyName*>)
Python Example	`activeDSO = oDesktop.GetRegistryString('Desktop/ActiveDSOConfigurations/HFSS 3D Layout')`

GetRunningInstancesMgr

Returns the object of the Running Instances Manager.

UI Access	N/A
Parameters	Name | Type | Description None
Return Value	Running instances manager object

Python Syntax	GetRunningInstancesMgr()
Python Example	`oRunningInstances = oDesktop.GetRunningInstancecMgr()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSchematicEnvironment

Returns the name of the current schematic environment set in **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**.

UI Access	N/A
Parameters	None.
Return Value	Integer representing a schematic environment: • 0 = Circuit • 1 = Twin Builder • 2 = Maxwell Circuit

Python Syntax	GetSchematicEnvironment()
Python Example	`oDesktop.GetSchematicEnvironment()`

GetScriptingToolsHelper

> **Note:**
> This command is for internal Ansys use only.

Returns the object for the scripting tools helper.

UI Access	NA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>None</td><td></td><td></td></tr></table>
Return Value	Object ScriptingTools helper object

Python Syntax	GetScriptingToolsHelper()
Python Example	`oDesktop.GetScriptingToolsHelper()`

GetSysLibDirectory

Get the path to the `SysLib` directory.

UI Access	N/A
Parameters	None.
Return Value	String path to the `SysLib` directory.

Python Syntax	GetSysLibDirectory()
Python Example	`oDesktop.GetSysLibDirectory()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetTempDirectory

Gets the path to the `Temp` directory.

UI Access	N/A
Parameters	None.
Return Value	String path to the `Temp` directory.

Python Syntax	GetTempDirectory()
Python Example	`oDesktop.GetTempDirectory()`

GetUserLibDirectory

Gets the path to the UserLib directory.

UI Access	N/A
Parameters	None.
Return Value	Stringpath to the `UserLib` directory.

Python Syntax	GetUserLibDirectory()
Python Example	`oDesktop.GetUserLibDirectory()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetVersion

Returns a string representing the version.

UI Access	N/A
Parameters	None.
Return Value	String containing version of the product.

Python Syntax	GetVersion()
Python Example	`oDesktop.GetVersion()`

IsFeatureEnabled

Returns a Boolean for whether a queried feature is enabled.

UI Access	N/A
Parameters	<FeatureID>.
Return Value	Boolean for named feature.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	IsFeatureEnabled()

Python Example	`import ScriptEnv`
	`ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")`
	`oDesktop.RestoreWindow()`
	`feature_strs = ["SF3519", "F195709_EXPORT_TO_EMIT", "F362235_EMIT_RESULTS_WINDOW_IMPROVEMENTS",_`
	`"F136736_SBR_Rough_Surface", "F353006_VOLUMETRIC_SBR", "F359673_SBR_MULTISTATE_ARRAY", "F393115_SWE_ANTENNA",_`
	`"S196592_SBR_Directivity", "F432541_VSBR_IMPROVEMENTS", "F353007_VRT_FILTERS_ENHANCE", "S540337_SBR_REGION_LOSS_DIRECTIVITY",_`
	`"F11941_VRT_CURRENT_DENSITY", "S544593_SBR_3D_COMPONENT_ARRAY", "F539850_SBR_GO_BLOCKAGE", "F540275_SBR_RAYSTATS"]`
	`results = [oDesktop.IsFeatureEnabled(str) for str in feature_strs]`
	`result_txt = open("C:/Users/MyResults/Downloads/results.txt", "w")`
	`for i in range(len(feature_strs)):`
	`result_txt.write('%s : %s\n' % (feature_strs[i], results[i]))`
	`result_txt.close()`

KeepDesktopResponsive

Specifies the minimum number of milliseconds to keep the desktop from showing hung.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name: *<MinTimeInMilliseconds>* — Type: Integer — Description: The minimum number of milliseconds to keep the desktop window from showing hung, that is, not responding.
Return Value	Boolean True for success and to keep running; False to indicate that the calling script should shut down.

Python Syntax	KeepDesktopResponsive (<*TimeInMilliseconds*>)
Python Example	`oDesktop.KeepDesktopResponsive(10000)`

LaunchJobMonitor

Starts job monitoring. This brings up the **Monitor Job** dialog box.

UI Access	**Tools** > **Job Management** > **Monitor Jobs**
Parameters	Name: *<projectPath>* — Type: String — Description: Path to the project file to be monitored
Return Value	None

Python Syntax	LaunchJobMonitor(<*projectPath*>)
Python Example	`oDesktop.LaunchJobMonitor("C:\\projects\\basic.aedt")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

NewProject

Creates a new project. The new project becomes the active project.

UI Access	**File** > **New**.
Parameters	None.
Return Value	Object, the project that is added.

Python Syntax	NewProject()
Python Example	`oProject = oDesktop.NewProject()`

OpenAndConvertProject

Opens a legacy project and converts or copies it to .aedt format.

UI Access	Click **File** > **Open**, and choose a legacy project
Parameters	

Name	Type	Description
<itemPath>	String	full project path of the legacy project
<legacyChoice>	Integer	0: show conversion dialog box, (same as **File** > **Open** of a legacy file) 1: rename (changes extension to .aedt, the original file and results are renamed) 2: copy (creates new file with .aedt extension, and the original file and results remain available)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	An object reference to the newly opened project which has the .aedt extension

> **Warning:**
>
> If project file/results with the same name and .aedt extension already exist in the same directory, they will be overwritten.

Python Syntax	OpenAndConvertProject(*filePath*, *legacyChoice*)

Python Example	`oProject = oDesktop.OpenAndConvertProject("c:\files\optimtee.hfss", 1)`
	Note: optimtee.hfss is gone after this code executes
	`oProject = oDesktop.OpenAndConvertProject("c:\files\optimtee.hfss", 2)`
	Note: optimtee.hfss remains after this code executes

OpenProject

Opens a specified project.

UI Access	Click **File** > **Open**.
Parameters	Name | Type | Description <*FileName*> | String | Full path of the project to open.
Return Value	An object reference to the newly opened project.

Python Syntax	OpenProject(<*FileName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesktop.OpenProject("D:/Projects/Project1.aedt")`

OpenProjectWithConversion

> **Note:**
> This command is for internal Ansys use only.

Python Syntax	OpenProjectWithConversion()
Python Example	`oDesktop.OpenProjectWithConversion()`

PageSetup (Layout Editor)

Use: Specifies page setup for printing.

Command: **File>Page Setup**

Syntax: PageSetup <ArgArray>

Return Value: None.

Parameters: `<Margins>: Page margins in implicit units of inches.`

<Border>: Integer value indicating to draw border (1) or not to draw (0).

<DesignVars>: Integer value indicating to draw design vars (1) or not to draw (0).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PauseRecording

Temporarily stop script recording.

UI Access	NA
Parameters	Name | Type | Description None
Return Value	None

Python Syntax	PauseRecording()
Python Example	`oDesktop.PauseRecording()`

PauseScript

Pause the execution of the script and pop up a message to the user. The script execution will not resume until the user chooses.

UI Access	**Tools** > **Pause Script**
Parameters	Name | Type | Description <*Message*> | String | Any Text.
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	PauseScript (<*Message*>)
Python Example	`oDesktop.PauseScript("Text to display in pop-up dialog box.")`

Print

Prints the contents of the active view window.

UI Access	**File** > **Print**.
Parameters	None.
Return Value	None.

Python Syntax	Print()
Python Example	`oDesktop.Print()`

QuitApplication

Exits the desktop.

UI Access	**File** > **Exit**.
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	QuitApplication()
Python Example	`oDesktop.QuitApplication()`

RefreshJobMonitor

For use in monitoring a job.

UI Access	**Tools** > **Job Management** > **Monitor Jobs**.
Parameters	None.
Return Value	A string specifing the job state. The result can be any of the following strings: • "Monitor Not Visible" • "Queued" • "Running" • "Shutting Down" • "Unknown" • "Completed" • "Not Monitoring" • "Starting Monitoring"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	RefreshJobMonitor()
Python Example	`oDesktop.RefreshJobMonitor()`

ResetLogging

Redirects simulation log file to a specified directory and log level.

UI Access	N/A
Parameters	Name | Type | Description *<logFile>* | String | Path to log file. *<logLevel>* | Integer | Specified log level.
Return Value	None.

Python Syntax	ResetLogging(*<logFile>*, *<logLevel>*)
Python Example	`oDesktop.ResetLogging("C:/Project1.aedtresults/", 1)`

RestoreProjectArchive

Restores a previously archived project to a specified path.

UI Access	**File** > **Restore Archive**.
Parameters	Name | Type | Description *<ArchiveFilePath>* | String | Path to archived file *<ProjectFilePath>* | String | Path to restore location

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<OverwriteExistingFiles>	Boolean	True to overwrite an existing file of the same name; else False.
	<OpenProjectAfterRestore>	Boolean	True to open the project after it is restored; else False.
Return Value	None.		

Python Syntax	RestoreProjectArchive (*<Archivefilepath>*, *<ProjectFilePath>*, *<OverwriteExistingFiles>*, *<OpenProjectAfterRestore>*)
Python Example	`oDesktop.RestoreProjectArchive("C:\Users\jdoe\Documents\OptimTee.aedtz", "C:\Documents\OptimTee.aedt", False, True)`

RestoreWindow

Restores a minimized Desktop window.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	RestoreWindow()
Python Example	`oDesktop.RestoreWindow()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ResumeRecording

Resume recording a script.

UI Access	N/A
Parameters	None.
Return Value	None

Python Syntax	ResumeRecording()
Python Example	`oDesktop.ResumeRecording()`

RunACTWizardScript

> **Note:**
> This command is for internal Ansys use only.

Python Syntax	RunACTWizardScript()
Python Example	`oDesktop.RunACTWizardScript()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RunProgram

Runs an external program.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ProgName*></td><td>String</td><td>Name of the program to run.</td></tr><tr><td><*ProgPath*></td><td>String</td><td>Location of the program. Pass in an empty string to use the system path.</td></tr><tr><td><*WorkPath*></td><td>String</td><td>Working directory in which program will start.</td></tr><tr><td><*ArgArray*></td><td>Array of Strings</td><td>Arguments to pass to the program. If no arguments, pass in <code>None</code></td></tr></table>
Return Value	None

Python Syntax	RunProgram (<*ProgName*>, <*ProgPath*>, <*WorkPath*>, <*ArgArray*>)
Python Example	`oDesktop.RunProgram("winword.exe", _` `"C:\Program Files\Microsoft Office\Office10","", None)`

RunScript

Launches another script from within the script currently being executed.

UI Access	**Tools** > **Run Script**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*SCriptPath*>	String	Name or full path of the script to execute. If the full path to the script is not specified, Twin Builder searches for the specified script in the following locations, in this order: 1. Personal library directory. This is the **PersonalLib** subdirectory in the project directory. The project directory can be specified in the **General Options** dialog box (click >**Tools** > **Options** > **General Options** to open this dialog box) under the **Project Options** tab. 2. User library directory. This is the **userlib** subdirectory in the library directory. The library directory can be specified **General Options** dialog in the box (click **Tools** > **Options** > **General Options** to to open this dialog box) under the **Project Options** tab. 3. System library directory. This is the syslib subdirectory in the library directory. The library directory can be specified in the **General Options** dialog box (click **Tools** > **Options** > **General Options** to open this dialog box) under the **Project Options** tab. 4. HFSS installation directory.
Return Value	The return code (in long format) for the script method.		

Python Syntax	RunScript (<*ScriptPath*>)
Python Example	`oDesktop.RunScript("C:/Project/test1.vbs")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RunScriptWithArguments

Similar to RunScript, the command launches another script from within the currently executing script, but with arguments.

<table>
<tr><td>UI Access</td><td>NA</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ScriptPath></td><td>String</td><td>The name or full path of the script to execute. If the full path to the script is not specified, the software looks for the script in the following locations:
Personal library directory: "PersonalLib"
The PersonalLib directory can be specified in Tools > Options > General Options on the 'Project Options' tab.
User library directory: "userlib"
The UserLib directory can be specified in Tools > Options > General Options on the 'Project Options' tab.
System library directory: "syslib"
The SysLib directory can be specified in Tools > Options > General Options on the 'Project Options' tab.
Software installation directory</td></tr>
<tr><td><Arguments></td><td>String</td><td>The arguments to supply to the specified script.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Return code (in long format) for the script method.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	RunScriptWithArguments (<*ScriptPath*>, <*Arguments*>)
Python Example	`oDesktop.RunScriptWithArguments ("C:/Project/test2.py", "foo")`

SelectScheduler

Selects the scheduler used for batch job submission. It tries non-graphical selection of the scheduler, attempting to get version information from the scheduler in order to check for successful selection. If unable to get the information, it displays the **Select Scheduler** window and waits for the user to complete the settings.

UI Access: **Tools** > **Job Management** > **Select Scheduler**.

Parameters

Name	Type	Description
<*option*>	String	One of the following options (not case sensitive): • Empty string for remote RSM service • "RSM" for local RSM • "Windows HPC" for Windows HPC • "LSF" for Load-Sharing Facility • "SGE" for Grid Engine (GE, OGE, SGE, UGE, etc.) • "PBS" for Portable Batch Scheduler/System (PBSPro, Torque, Maui, etc.) • "Ansys Cloud" for Ansys Cloud
<*address (optional)*>	String	String specifying the IP address or hostname of the head node or for the remote host running the RSM service.
<*username (optional)*>	String	Username string to use for remote RSM service (or blank to use username stored in current submission host user settings). If the (non-blank) username doesn't match the username stored in current submission host user settings, then the Select Scheduler dialog is displayed to allow for pass-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			word entry prior to job submission.
	<forcePasswordEntry (optional)>	String	Boolean used to force display of the Select Scheduler GUI to allow for password entry prior to job submission.
Return Value	The selected scheduler (if selection was successful, this string should match the input option string, although it could differ in upper/lowercase).		

Python Syntax	Select Scheduler(*<option>, <address>, <username>, <forcePasswordEntry>*)
Python Example	`result = oDesktop.SelectScheduler("Windows HPC",` `"headnode.win.example.com")`

SetActiveProject

Specifies the name of the project that should become active in the desktop. Returns that project.

UI Access	N/A		
Parameters	Name	Type	Description
	<ProjectName>	String	The name of the project already in the Desktop that is to be activated.
Return Value	Object, the project that is activated.		

Python Syntax	SetActiveProject (*<ProjectName>*)
Python Example	`oProject = oDesktop.SetActiveProject("Project1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetActiveProjectByPath

Specifies the name of the project that should become active in the desktop. Returns that project. If a user has two projects open with the same name, the result of `SetActiveProject` is ambiguous (the first one listed in selected). This command permits unambiguous specification of the active project.

UI Access	N/A
Parameters	Name | Type | Description <*ProjectName*> | String | The full path name of the project already in the Desktop that is to be activated.
Return Value	Object, the project that is activated.

Python Syntax	SetActiveProjectByPath(<*ProjectName*>)
Python Example	`oProject = oDesktop.SetActiveProjectByPath("c:\Projects\MyProject.aedt")`

SetCustomMenuSet

Sets the custom menu set for Electronics Desktop.

UI Access	Navigate to **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**. Select a configuration using the **Custom Menu Set** drop-down menu.
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*customMenuSet*>	String	Name of desired menu set. Can be one of: • 'Default' • 'EM' • 'RF' • 'RF.0' • 'SI' • 'SI1.0' • 'SI2.0' • 'Twin Builder'
Return Value	None		

Python Syntax	SetCustomMenuSet(<*customMenuSet*>)
Python Example	`oDesktop.SetCustomMenuSet('Default')`

SetDesktopConfiguration

Sets the desktop configuration.

UI Access	Navigate to **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**. Select a configuration using the **Set targeted configuration** drop-down menu.		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*configName*>	String	Name of desired Desktop configuration. Can be one of: • 'All' • 'EM' • 'RF' • 'SI' • 'Twin Builder'
Return Value	None		

Python Syntax	SetDesktopConfiguration (<*configName*>)
Python Example	`oDesktop.SetDesktopConfiguration('RF')`

SetLibraryDirectory

Sets the library directory path. The specified directory must already exist and contain a syslib folder.

UI Access	NA		
Parameters	Name	Type	Description
	<*DirectoryPath*>	String	The path to the SysLib Directory
Return Value	None		

Python Syntax	SetLibraryDirectory (<*DirectoryPath*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesktop.SetLibraryDirectory("c:\libraries")`

SetProjectDirectory

Sets the project directory path.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DirectoryPath*></td><td>String</td><td>The path to the project directory. This should be writeable by the user.</td></tr></table>
Return Value	None.

Python Syntax	SetProjectDirectory (<*DirectoryPath*>)
Python Example	`oDesktop.SetProjectDirectory("c:\projects")`

SetRegistryFromFile

Configures registry by specifying an Analysis Configuration file, which must have been exported from the HPC and Analysis panel.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*filePath*></td><td>String</td><td>Full file path of registry file.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Success if configuration is imported. Bad argument value if the file is not found or does not contain valid analysis configuration data.

Python Syntax	SetRegistryFromFile(<*filePath*>)
Python Example	`oDesktop.SetRegistryFromFile('c:/temp/test.acf')`

SetRegistryInt

Sets registry key to an integer value.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path. <*int*> | Integer | Integer value to be assigned to registry key.
Return Value	Success if the key is defined as an integer. Bad argument value if a key is not defined, or if the value is a text string.

Python Syntax	SetRegistryInt(<*KeyName*>, <*int*>)
Python Example	`oDesktop.SetRegistryInt('Desktop/Settings/ProjectOptions/HFSS 3D Layout/UpdateReportsDynamicallyOnEdits', 0)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetRegistryString

Sets registry key to a string value.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path. <*value*> | String | String value to be assigned to registry key.
Return Value	Success if the key is defined as a text string. Bad argument value if the key is not defined or requires an integer value.

Python Syntax	SetRegistryString(<*KeyName*>, <*value*>)
Python Example	`oDesktop.SetRegistryString('Desktop/ActiveDSOConfigurations/HFSS 3D Layout', 'Local')`

SetSchematicEnvironment

Sets the schematic environment for Electronics Desktop.

UI Access	Navigate to **Tools** > **Options** > **General Options** > **General** > **Desktop Configuration**. Select a schematic environment using the **Custom Menu Set** drop-down menu.
Parameters	Name | Type | Description <*schEnv*> | Integer | Desired schematic environment. Can be one of:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• 0 (Circuit) • 1 (Twin Builder) • 2 (Maxwell Circuit)
Return Value	None		

Python Syntax	SetSchematicEnvironment(<*schEnv*>)
Python Example	`oDesktop.SetSchematicEnvironment(1)`

SetTempDirectory

Sets the temp directory path. The directory will be automatically created if it does not already exist.

UI Access	N/A		
Parameters	Name	Type	Description
	<*DirectoryPath*>	String	The path to the Temp directory. This should be writeable by the user.
Return Value	None.		

Python Syntax	SetTempDirectory (<*DirectoryPath*>)
Python Example	`oDesktop.SetTempDirectory("c:\tmp")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ShowDockingWindow

Shows or hides a docking window.

UI Access	Right click docking window > Show/Hide.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*windowName*></td><td>String</td><td>The window name (for example, "Message Manager", "Component Libraries", "Properties")</td></tr><tr><td><*show*></td><td>Boolean</td><td>True to show; False to hide.</td></tr></table>
Return Value	None.

Python Syntax	ShowDockingWindow (<*windowName*>, <show>)
Python Example	`oDesktop.ShowDockingWindow('Message Manager',False)`

Sleep

Suspends execution of the solver for the specified number of milliseconds, up to 60,000 milliseconds (1 minute).

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*TimeInMil-liseconds*></td><td>Integer</td><td>The time that the execution should be suspended in milliseconds</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	Sleep (<*TimeInMilliseconds*>)
Python Example	`oDesktop.Sleep(1000)`

StopSimulations

Either cleanly stops all running and pending simulations, or aborts them.

UI Access	NA
Parameters	Name: <bool> <Clean> — Type: Integer — Description: If true, clean stop for all running and pending simulations. If false, aborts them.
Return Value	A human readable status string specifying what happened.

Python Syntax	StopSimulations (<bool>, clean)
Python Example	`oDesktop.StopSimulations(bool clean)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SubmitJob

Submits a batch job to a scheduler. When submitting the same project file multiple times, you should have the script wait for each job (or jobs for multi-step) to finish, which can be done via the monitoring functions LaunchJobMonitor() and RefreshJobMonitor(), checking the result of RefreshJobMonitor() in a loop until it returns completed ("Monitor Not Visible") status.

UI Access	**Tools** > **Job Management** > **Submit Job**.
Parameters	

Name	Type	Description
<settingsPath>	String	Path to the settings file (exported from the Submit Job dialog box) to use for submission.
<projectPath>	String	Path to the project file to use in the batch job. This could be an archive (.aedtz file) or an un-archived project.
<design>(optional)	String	Name of the design to use for batch solve.
<setup>(optional)	String	Name of the specific setup to solve.

Return Value	Array of job ID strings (empty if no jobs submitted).

Python Syntax	SubmitJob(*<settingsPath>*, *<projectPath>*, *<design>*, *<setup>*)

Python Example	`jobIDs = oDesktop.SubmitJob("C:\\hpc-settings\\Submit_Job_Settings.areg", "C:\\projects\\basic.aedt"))`
	`moreIDs = oDesktop.SubmitJob("C:\\hpc-settings\\Submit_Job_Settings.areg", "C:\\projects\\basic.aedt", "Design1", "Setup1")`
	With an RSM scheduler with LSDSO:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oDesktop.SubmitJob("C:\\Test\\Submit_Job_Settings_LSDSO.areg", "c:\\test\\test1.aedtz", "", "HFSSDesign1:Optimetrics:ParametricSetup1")`

TileWindows

Arranges all open windows in a tiled format.

UI Access	From main menu, **Window** >**Tile Horizontally** or **Window** >**Tile Vertically**.
Parameters	Name: <*TilingFlag*> — Type: Integer — Description: • 0 – Tile vertically. • 1 – Tile horizontally.
Return Value	None.

Python Syntax	TileWindows(<*TilingFlag*>)
Python Example	`oDesktop.TileWindows(0)`

Desktop Commands For Registry Values

The Ansys Registry is stored as XML format file. By default it is located at C:\Users\<UserName>\Documents\Ansoft\<AnsysProductNameversion>\config\<PC_NAME>_user.XML. Most of the Ansys product configuration information is stored in this XML file. These methods allow you to change the product configuration.

For example, to set the DSO & HPC analysis setup for HFSS using a Python script:

1. Start HFSS 3D Layout.
2. Go to the DSO and HPC options and create a setup named "test".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

3. Export the setup to a file (for example, c:\temp\test.acf).
4. Copy the exported file to a target PC (for example, f:\temp\test.acf).
5. Run the following script:

```
#import the setup

oDesktop.SetRegistryFromFile("f:\\temp\\test.acf")

# Set Active Setup to "test"

oDesktop.SetRegistryString("Desktop/ActiveDSOConfigurations/HFSS 3D Layout", "test")
```

See the following subtopics:

DeleteRegistryEntry

DoesRegistryValueExist

GetRegistryInt

GetRegistryString

SetRegistryFromFile

SetRegistryInt

SetRegistryString

DeleteRegistryEntry

Deletes a registry entry from a registry key. Returns true if deletion succeeded.

UI Access	N/A		
Parameters	Name	Type	Description
	<pathToRegistry>	String	Path to Registry entry.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Boolean: • **True** – Key has been deleted. • **False** – Key does not exist.

Python Syntax	DeleteRegistryEntry(<*pathToRegistry*>)
Python Example	`res = oDesktop.DeleteRegistryEntry("Desktop/ColorScheme")`

DoesRegistryValueExist

Determines whether a registry value exists.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path.
Return Value	Boolean: • **True** – Key exists. • **False** – Key does not exist.

Python Syntax	DoesRegistryValueExist(<*KeyName*>)
Python Example	`Exist = oDesktop.DoesRegistryValueExist('Desktop/ActiveDSOConfigurations/HFSS 3D Layout')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetRegistryInt

Obtains registry key integer value.

UI Access	N/A
Parameters	Name | Type | Description *<KeyName>* | String | Full name of registry key, including path.
Return Value	Integer if the integer value is found. Return as Bad-Argument-Value if registry key does not exist or it is not an integer value.

Python Syntax	GetRegistryInt(<*KeyName*>)
Python Example	`num = oDesktop.GetRegistryInt('Desktop/Settings/ProjectOptions/HFSS 3D Layout/UpdateReportsDynamicallyOnEdits')`

GetRegistryString

Obtains registry key string value.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*KeyName*></td><td>String</td><td>Full name of registry key, including path.</td></tr></table>
Return Value	String if the string value is found. Return as Bad-Argument-Value if registry key does not exist or it is not a string value.

Python Syntax	GetRegistryString(<*KeyName*>)
Python Example	`activeDSO = oDesktop.GetRegistryString('Desktop/ActiveDSOConfigurations/HFSS 3D Layout')`

SetRegistryFromFile

Configures registry by specifying an Analysis Configuration file, which must have been exported from the HPC and Analysis panel.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*filePath*></td><td>String</td><td>Full file path of registry file.</td></tr></table>
Return Value	Success if configuration is imported. Bad argument value if the file is not found or does not contain valid analysis configuration data.

Python Syntax	SetRegistryFromFile(<*filePath*>)
Python Example	`oDesktop.SetRegistryFromFile('c:/temp/test.acf')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetRegistryInt

Sets registry key to an integer value.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path. <*int*> | Integer | Integer value to be assigned to registry key.
Return Value	Success if the key is defined as an integer. Bad argument value if a key is not defined, or if the value is a text string.

Python Syntax	SetRegistryInt(<*KeyName*>, <*int*>)
Python Example	`oDesktop.SetRegistryInt('Desktop/Settings/ProjectOptions/HFSS 3D Layout/UpdateReportsDynamicallyOnEdits', 0)`

SetRegistryString

Sets registry key to a string value.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<value>	String	String value to be assigned to registry key.
Return Value	Success if the key is defined as a text string. Bad argument value if the key is not defined or requires an integer value.		

Python Syntax	SetRegistryString(*<KeyName>*, *<value>*)
Python Example	`oDesktop.SetRegistryString('Desktop/ActiveDSOConfigurations/HFSS 3D Layout', 'Local')`

ImportExport Tool Commands

These oDesktop commands are run by using the GetTool script to call the ImportExport Tool.

```
oTool = oDesktop.GetTool("ImportExport")
```

Scripts run via the ImportExport Tool include:

ImportANF

ImportANFV2

ImportAutoCAD

ImportAWRMicrowaveOffice

ImportEDB

ImportExtracta

ImportGDSII

ImportGerber

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportIDF

ImportIDFandMerge

ImportIPC

ImportODB

ImportXFL

ImportANF

Imports an ANF file into a new project. For older ANFv2 files, use ImportANFv2.

UI Access	**File** > **Import** > **ANF**.		
Parameters	Name	Type	Description
	<ANFfilename>	String	Full path of ANF file.
Return Value	None.		

Python Syntax	ImportANF(*<ANFfilename>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportANF('C:\\AnsTranslator\\results\\package4.anf')`

ImportANFv2

Imports an ANFv2 file into a new project. For newer ANF files, use ImportANF.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**File** > **Import** > **ANF**.		
Parameters	Name	Type	Description
	<ANFfilename>	String	Full path of ANF file.
	<outputPathName>	String	Full path of *.aedb output file.
	<controlFileName>	String	Full path of XML control file.
	<cmpFileName>	String	Full path of CMP file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportANFv2 (*<ANFfilename>*, *<outputPathName>*, *<controlFileName>*, *<cmpFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportANFV2("C:/Users/jdoe/Documents/SAMPLEFILES/my_model.anf",`
	`"C:/Users/jdoe/Documents/Ansoft/my_model.aedb",`
	`"C:/Users/jdoe/Documents/Ansoft/my_model.xml", "")`

ImportAutoCAD

Imports an AutoCAD file into a new project.

UI Access	**File** > **Import** > **AutoCAD**.		
Parameters	Name	Type	Description
	<dxfFileName>	String	Full path of DXF file.
	<outputPathFileName>	String	Full path of EDB file to create during import.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*controlFileName*>	String	Full path of XML control file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportAutoCAD (<*dxfFileName*>, <*outputPathFileName*>, <*controlFileName*>)
Python Example	`oTool = oDesktop.GetTool('ImportExport')` `oTool.ImportAutoCAD('C:/MyPath/a4lines.dxf', 'C:/MyPath/a4lines.aedb.edb', 'C:/MyPath/a4lines.xml')`

ImportAWRMicrowaveOffice

Imports an AWRMicrowaveOffice file into a new project.

UI Access	**File** > **Import** > **AWRMicrowaveOffice**.		
Parameters	Name	Type	Description
	<*xmlFileName*>	String	Full path of XML file.
	<*outputPathName*>	String	Full path of output file.
	<*logFileName*>	String	Full path of log file.
Return Value	None.		

Python Syntax	ImportAWRMicrowaveOffice (<*xmlFileName*>, <*outputPathName*>, <*logFileName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportAWRMicrowaveOffice('C:/MyFiles/package4.xml', 'C:/MyFiles/package4.aedb', 'C:/MyFiles/package4.log')`

ImportEDB

Imports an EDB file into a new project.

UI Access	**File** > **Import** > **EDB**.
Parameters	Name | Type | Description *<edbFileName>* | String | Full path of EDB file.
Return Value	None.

Python Syntax	ImportEDB (*<edbFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportEDB('C:/MyFiles/projectforimport.aedb/edb.def')`

ImportExtracta

Imports a Cadence Extracta file into a new project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

In order for this script to work, you must have the Cadence-supplied executable Extracta.exe installed on your machine.

UI Access	**File** > **Import** > **Cadence APD / Allegro / SiP**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*extractaFileName*></td><td>String</td><td>Full path of Extracta file.</td></tr><tr><td><*outputPathName*></td><td>String</td><td>Full path of output file to be created.</td></tr><tr><td><*controlFileName*></td><td>String</td><td>Optional. Full path to XML control file.</td></tr></table>
Return Value	None.

Python Syntax	ImportExtracta (<*extractaFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportExtracta('C:/MyFiles/projectforimport.brd', 'C:/MyFiles/project.edb', '')`

ImportGDSII

Imports a GDSII file into a new project.

UI Access	**File** > **Import** > **GDSII**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*gdsiiFileName*></td><td>String</td><td>Full path of GDSII file.</td></tr><tr><td><*outputPathName*></td><td>String</td><td>Full path of EDB file to create during import.</td></tr><tr><td><*controlFileName*></td><td>String</td><td>Optional. Full path of XML control file. Full path of “technology” (corresponding to `-t=technologyfile` argument in anstranslator). Pass empty string if none.</td></tr><tr><td><*mapFileName*></td><td>String</td><td>If technology file was used in place of the control file. Full path to either gdsii mapping file (corresponding to `-g=gdsMappingFile` argument in anstranslator) or XML control file. Otherwise, pass empty string.</td></tr></table>
Return Value	None.

Python Syntax	ImportGDSII(<*gdsiiFileName*>, <*outputPathName*>, <*controlFileName*>, <*mapFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportGDSII('C:/Files/test.gds', 'C:/Files/test.aedb',`
	`'C:/Files/test.ircx', 'C:/Files/test.xml')`

ImportGerber

Imports a GERBER file into a new project.

UI Access	**File** > **Import** > **GERBER**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*gerberFileName*></td><td>String</td><td>Full path of GERBER file.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<outputPathName>	String	Full path of EDB file to create during import.
	<controlFileName>	String	Optional. Full path of XML control file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportGerber(*<gerberFileName>*, *<outputPathName>*, *<controlFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportGERBER('C:/Files/test.gbr', 'C:/Files/test.aedb.edb',`
	`'C:/Files/test.xml')`

ImportIDF

Imports an IDF file into a new project. To merge with an existing design, use ImportIDFandMerge.

UI Access	placeholder		
Parameters	Name	Type	Description
	<NAME>	string	Settings
	<Board>	bool	Full path of board file
	<Library>	int	Full path of library file
	<Control>	int	Full path of control file
	<Filters>	list	["Cap","Height","HeightExclude2D","Ind","Power","Res"]
	<CreateFilteredAsNonModel>	bool	True or False

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<FootPrint>	string	Footprint size and unit
	<NAME>	string	definitionOverridesMap
	<NAME>	string	instanceOverridesMap
	<HighSurfThickness>	string	High side surface thickness and unit
	<LowSurfThickness>	string	Low side surface thickness and unit
	<InternalLayerThickness>	string	Internal layer thickness and unit
	<NumInternalLayer>	int	Number of internal layers
	<HighSurfaceCopper>	int	High side surface coverage percentage
	<LowSurfaceCopper>	int	Low side surface coverage percentage
	<InternalLayerCopper>	int	Internal layer coverage percentage
	<TraceMaterial>	string	Trace material name
	<SubstrateMaterial>	int	Substrate material name
	<CreateBoard>	bool	True or False
	<ModelBoardAsRect>	bool	True or False
	<ModelDeviceAsRect>	bool	True or False
	<Cutoff>	bool	True or False
	<IncludeDrilledHoles>	bool	True or False
	<HoleDiameterCutoff>	string	Hole diameter size and unit
	<ReplaceDevices>	bool	True or False
	<CreatePointsOnBoardForInstances>	bool	True or False
Return Value	None.		

Python Syntax	ImportIDF_1 (*<NAME>*, *<Board>*, *<Library>*, *<Control>*, *<Filters>*, *<CreateFilteredAsNonModel>*, *<NAME>*, *<NAME>*, *<HighSurfThickness>*, *<LowSurfThickness>*, *<InternalLayerThickness>*, *<NumInternalLayer>*, *<HighSurfaceCopper>*, *<LowSurfaceCopper>*, *<InternalLayerCopper>*, *<TraceMaterial>*, *<SubstrateMaterial>*, *<CreateBoard>*, *<ModelBoardAsRect>*, *<ModelDeviceAsRect>*, *<Cutoff>*, *<IncludeDrilledHoles>*, *<ReplaceDevices>*)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example

```
oDesign.ImportIDF(
	[
		"NAME:Settings",
		"Board:="		, "C:\\Users\\Model Files\\brd_board.emn",
		"Library:="		, "C:\\Users\\Model Files\\brd_board.emp",
		"Control:="		, "C:\\Users\\Model Files\\brd_board.xml",
		"Filters:="		, ["HeightExclude2D"],
		"CreateFilteredAsNonModel:=", False,
		"FootPrint:="		, "0.1mm2",
		[
			"NAME:definitionOverridesMap"
		],
		[
			"NAME:instanceOverridesMap"
		],
		"HighSurfThickness:="	, "0.07mm",
		"LowSurfThickness:="	, "0.07mm",
		"InternalLayerThickness:=", "0.07mm",
		"NumInternalLayer:="	, 2,
		"HighSurfaceCopper:="	, 30,
		"LowSurfaceCopper:="	, 30,
		"InternalLayerCopper:="	, 30,
		"TraceMaterial:="	, "Cu-Pure",
		"SubstrateMaterial:="	, "FR-4",
		"CreateBoard:="		, True,
		"ModelBoardAsRect:="	, True,
		"ModelDeviceAsRect:="	, False,
		"Cutoff:="		, False,
		"IncludeDrilledHoles:="	, True,
		"HoleDiameterCutoff:="	, "0.1mm",
		"ReplaceDevices:="	, False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"CreatePointsOnBoardForInstances:=", False])

ImportIDFandMerge

Imports an IDF file into a new project. To import without merging, use ImportIDF.

UI Access	**File** > **Import** > **IDF**. Enable **Merge with existing** and select a project/design.
Parameters	See table below
Return Value	None.

Name	Type	Description
<idfFileName>	String	Full path of GERBER file.
<libFileName>	String	Optional. Full path of library file. Pass empty string if none.
<controlFileName>	String	Optional. Full path of XML control file. Pass empty string if none.
<project>	String	Name of project to merge with.
<design>	String	Name of design to merge with.

Python Syntax	ImportIDFandMerge(*<idfFileName>*, *<libPathName>*, *<controlFileName>*, *<project>*, *<design>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportIDFandMerge('C:/Files/test.brd', 'C:/Files/test.aedb.lib',`
	`'C:/Files/test.xml', 'Project1', 'Design12')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportIDX

Imports and IDX model into a new Icepak project.

UI Access	**Icepak** > **Import IDX**

Parameters

Name	Type	Description
<NAME>	string	Settings
<Board>	bool	Full path of .idx file
<Library>	string	NA
<Control>	string	NA
<Filters>	list	HeightExclude2D
<CreateFilteredAsNonModel>	bool	True or False
<NAME>	string	definitionOverridesMap
<NAME>	string	instanceOverridesMap
<HighSurfThickness>	string	High surface layer thickness value and unit
<LowSurfThickness>	string	Low surface layer thickness value and unit
<InternalLayerThickness>	string	Internal surface layer thickness value and unit
<NumInternalLayer>	int	Number of internal layers value
<HighSurfaceCopper>	int	High surface percent coverage value
<LowSurfaceCopper>	int	Low surface percent coverage value
<InternalLayerCopper>	int	Internal layer percent coverage value
<TraceMaterial>	string	Trace material name
<SubstrateMaterial>	int	Substrate material name
<CreateBoard>	bool	True or False
<ModelBoardAsRect>	bool	True or False
<ModelDeviceAsRect>	bool	True or False
<Cutoff>	bool	True or False
<ReplaceDevices>	bool	True or False

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	ImportIDX (<*NAME*>, <*Board*>, <*Library*>, <*Control*>, <*Filters*>, <*CreateFilteredAsNonModel*>, <*NAME*>, <*NAME*>, <*HighSurfThickness*>, <*LowSurfThickness*>, <*InternalLayerThickness*>, <*NumInternalLayer*>, <*HighSurfaceCopper*>, <*LowSurfaceCopper*>, <*InternalLayerCopper*>, <*TraceMaterial*>, <*SubstrateMaterial*>, <*CreateBoard*>, <*ModelBoardAsRect*>, <*ModelDeviceAsRect*>, <*Cutoff*>, <*ReplaceDevices*>)

Python Example	`oDesign.ImportIDX(`
	`[`
	`"NAME:Settings",`
	`"Board:=" , "C:\\Users\\Model files\\PCB-00278_A.idx",`
	`"Library:=" , "",`
	`"Control:=" , "",`
	`"Filters:=" , ["HeightExclude2D"],`
	`"CreateFilteredAsNonModel:=", False,`
	`[`
	`"NAME:definitionOverridesMap"`
	`],`
	`[`
	`"NAME:instanceOverridesMap"`
	`],`
	`"HighSurfThickness:=" , "0.07mm",`
	`"LowSurfThickness:=" , "0.07mm",`
	`"InternalLayerThickness:=", "0.07mm",`
	`"NumInternalLayer:=" , 2,`
	`"HighSurfaceCopper:=" , 30,`
	`"LowSurfaceCopper:=" , 30,`
	`"InternalLayerCopper:=" , 30,`
	`"TraceMaterial:=" , "Cu-Pure",`
	`"SubstrateMaterial:=" , "FR-4",`
	`"CreateBoard:=" , True,`
	`"ModelBoardAsRect:=" , False,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "ModelDeviceAsRect:="   , False,
                "Cutoff:="              , False,
                "ReplaceDevices:="      , False
        ])
```

ImportIPC

Imports an IPC2581 file into a new project.

UI Access	**File** > **Import** > **IPC2581**.
Parameters	

Name	Type	Description
<*ipcFileName*>	String	Full path of IPC2581 file.
<*outputPathName*>	String	Full path of EDB file to create during import.
<*controlFileName*>	String	Optional. Full path of XML control file. Pass empty string if none.

Return Value	None.

Python Syntax	ImportIPC(<*ipcFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportIPC('C:/Files/test.cvg', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml', 'C:/Files/test.txt')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportODB

Imports an ODB++ file into a new project.

UI Access	**File** > **Import** > **ODB++**.
Parameters	

Name	Type	Description
<*odbFileName*>	String	Full path of ODB++ file.
<*outputPathName*>	String	Full path of EDB file to create during import.
<*controlFileName*>	String	Optional. Full path of XML control file. Pass empty string if none.

Return Value	None.

Python Syntax	ImportODB(<*odbFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportODB('C:/Files/test.odb', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml')`

ImportXFL

Imports an XFL file into a new project.

UI Access	**File** > **Import** > **XFL**
Parameters	

Name	Type	Description
<*xflFileName*>	String	Full path of XFL file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<outputPathName>	String	Full path of EDB file to create during import.
	<controlFileName>	String	Optional. Full path of XML control file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportXFL(*<xflFileName>*, *<outputPathName>*, *<controlFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportXFL('C:/Files/test.xfl', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

5 - Running Instances Manager Script Commands

The Running Instances Manager is a scripting object that lets you identify and connect to all running instances of Electronics Desktop. oDesktop objects that are returned provide full scripting functionality. Running Instances Manager commands should be executed by the oDesktop object. For example:

```
oRunningInstances = oDesktop.GetRunningInstancesMgr()
```

GetAllRunningInstances

GetRunningInstanceByProcessID

GetRunningInstanceByProject

GetAllRunningInstances

Returns a list of running instances of Ansys Electronics Desktop.

UI Access	N/A
Parameters	None.
Return Value	Array containing list of Ansys Electronics Desktop instances.

Python Syntax	GetAllRunningInstances()
Python Example	`obj = oRunningInstances.GetAllRunningInstances()`

GetRunningInstanceByProcessID

Returns the instance of Ansys Electronics Desktop that is running a specified process.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><processID></td><td>Integer</td><td>Process ID</td></tr></table>
Return Value	String of the returned instance.

Python Syntax	GetRunningInstanceByProcessID(<processID>)
Python Example	`obj = oRunningInstances.GetRunningInstanceByProcessID(12345)`

GetRunningInstancesMgr

Returns the object of the Running Instances Manager.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>None</td><td></td><td></td></tr></table>
Return Value	Running instances manager object

Python Syntax	GetRunningInstancesMgr()
Python Example	`oRunningInstances = oDesktop.GetRunningInstancecMgr()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

6 - Project Object Script Commands

Project commands should be executed by the oProject object.

One example of accessing this object is:

```
oProject = oDesktop.GetActiveProject()
```

Dataset Script Commands:

AddDataset

DeleteDataset

EditDataset

ExportDataset

HasDataset

ImportDataset

Other Project Object Script Commands:

AddMaterial

AnalyzeAll

ChangeProperty

ClearMessages

CloneMaterial

Close

CopyDesign

CutDesign

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteDesign

DeleteToolObject

EditMaterial

ExportMaterial

GetActiveDesign

GetArrayVariables

GetChildNames [Project]

GetChildObject [Project]

GetChildTypes [Project]

GetDefinitionManager

GetDependentFiles

GetDesign

GetDesigns

GetEDBHandle

GetLegacyName

GetName [Project]

GetObjPath [Project]

GetPath

GetPropEvaluatedValue

GetPropNames [Project]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropSIValue

GetPropValue [Project]

GetProperties

GetPropertyValue

GetTopDesignList

GetVariableValue

GetVariables

InsertDesign

InsertDesign (Layout Editor)

InsertDesignWithWorkflow

InsertToolObject

Paste [Project Object]

Redo [Project Level]

RemoveAllUnusedDefinitions

RemoveMaterial

RemoveUnusedDefinitions

Rename

RunToolkit

Save

SaveAs

SaveAsStandAloneProject

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SaveProjectArchive

SetActiveDefinitionEditor

SetActiveDesign

SetPropValue [Project]

SetPropertyValue

SetVariableValue

SimulateAll

Undo [Project]

UpdateDefinitions

AddDataset

Adds a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Add**		
Parameters	Name	Type	Description
	<*DatasetDataArray*>	Array	["NAME:<DatasetName>", ["NAME:Coordinates", <CoordinateArray>, <CoordinateArray>, ...]
	<*DatasetName*>	String	Name of the dataset.
	<*CoordinateArray*>	Array	["NAME:Coordinate", "X:=", <double>, "Y:=",<double>]
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddDataset <*DatasetDataArray*>
Python Example	oProject.AddDataset(["NAME:$ds1", ["NAME:Coordinates", ["NAME:Coordinate", "X:=", 2, "Y:=", 4], ["NAME:Coordinate", "X:=", 6, "Y:=", 8]]]) oDesign.AddDataset(["NAME:$ds1", ["NAME:Coordinates", ["NAME:Coordinate", "X:=", 2, "Y:=", 4], ["NAME:Coordinate", "X:=", 6, "Y:=", 8]]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

AddMaterial

Adds a local material.

UI Access	**Add Material** in the material editor.		
Parameters	Name	Type	Description
	<*MaterialParams*>	Array	["NAME: <name of the material to be added>", <MatProperty>, <MatProperty>, ...]
	<*MatProperty*>	Array	For simple material: "<PropertyName>:=", <value> For anisotropic material: ["NAME:<PropertyName>", "property_type:=", "AnisoProperty", "unit:=", <Unit>",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		"component1:=", <value>, "component2:=", <value>, "component3:=", <value>))]
<*PropertyName*>	String	Should be one of the following (depending on the material, design, and solution types): Electromagnetic (Maxwell-exclusive material properties omitted, see Maxwell Scripting help): "permittivity", "permeability", "conductivity", "dielectric_loss_tangent", "magnetic_loss_tangent", "electric_coercivity", "magnetic_coercivity", "saturation_mag", "lande_g_factor", "delta_H", "delta_h_freq", "mass_density" Thermal (including solids, Icepak fluid flow, and IcepakFEA rotating fluid modeling): "thermal_conductivity", "mass_density", "specific_heat", "thermal_expansion_coefficient", "thermal_material_type", "viscosity", "diffusivity", "molecular_mass", "clarity_type" Structural: "mass_density", "youngs_modulus", "poissons_ratio", "thermal_expansion_coefficient"
<*Unit*>	String	Possible values (Maxwell-exclusive properties omitted, see Maxwell Scripting Help; other missing entries are unitless):

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			conductivity: "siemens/m" saturation_mag: "uTesla", "mTesla", "tesla", "kTesla", "uGauss", "mGauss", "gauss", "kGauss" delta_H: "A_per_meter", "kA_per_meter", "Oe", "kOe" delta_h_frequency: "Hz", "kHz", "MHz", "GHz", "THz", "rps", "per_sec" mass_density: "kg/m^3" thermal_conductivity: "W/m-C" specific_heat: "J/kg-C" youngs_modulus: "N/m^2" thermal_expansion_coefficient: "1/C"
Return Value	None		

Python Syntax	AddMaterial (["NAME:<MaterialName>", <MatProperty>, <MatProperty>, ...])

Python Example	`oDefinitionManager.AddMaterial(`
	`["permittivity:=", "2.2", "0.002"])`
	
	`oDefinitionManager.AddMaterial(`
	`["NAME:Material2",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"dielectric_loss_tangent:=", "44", ["NAME:saturation_mag", "property_type:=", "AnisoProperty", "unit:=", "Gauss", "component1:=", "11", "component2:=", "22", "component3:=", "33"], "delta_H:=", "44Oe"])

AnalyzeAll [project]

Runs the project-level script command from the script, which simulates all solution setups and Optimetrics setups for all design instances in the project. The UI waits until simulation is finished before continuing with the script.

UI Access	**Project** > **Analyze All**
Parameters	None.
Return Value	None.

Python Syntax	AnalyzeAll()
Python Example	oProject.AnalyzeAll()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeProperty

Changes the properties of an object in the history tree.

<table>
<tr><td>UI Access</td><td>Right-click an object in the History Tree and select Properties.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><propertyArgs></td><td>Array</td><td>Structured array. The properties vary depending on the object. Due to the number of potential configurations, it is recommended that you generate this script using the UI's Automation tab.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	**Example: Changing the Position of a Box**
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DCmdTab",`
	`  ["NAME:PropServers" , "Box1:CreateBox:1" ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Position", "X:=" , "0.35in", "Y:=" , "0.55in", "Z:=" , "0in"`
	`  ]`
	`  ]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ]
  ])
```

Example: Offset the Position of a 3D Component or Layout Component with a Variable

```
oDesign.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:LocalVariableTab",
      ["NAME:PropServers",
        "LocalVariables"
      ],
      ["NAME:NewProps",
        ["NAME:zH",
        "PropType:=" , "VariableProp",
        "UserDef:=" , True,
        "Value:=" , "50mm"
        ]
      ]
    ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers",
        "LC1_1"
      ],
      ["NAME:ChangedProps",
        ["NAME:Position",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "X:=" , "0mm",
          "Y:=" , "0mm",
          "Z:=" , "zH"
          ]
       ]
     ]
  ])
```

Example: Changing a Box's Material and Wireframe Display

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers" , "Box1" ],
       ["NAME:ChangedProps",
       ["NAME:Material", "Value:=" , "\"vacuum\"" ],
       ["NAME:Display Wireframe", "Value:=" , True ]
      ]
    ]
  ])
```

Example: Change Default Handle (reference coordinate system) for a 3D Component or Layout Component

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers","LC1_1"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``
	        ["NAME:ChangedProps",
	          ["NAME:Handle","Value:=" , " NotchOutY"]
	        ]
	      ]
	   ])
	
	oEditor.ChangeProperty(
	   ["NAME:AllTabs",
	      ["NAME:Maxwell2D",
	        ["NAME:PropServers", "Design Settings"],
	        ["NAME:ChangedProps",
	          ["NAME:Model settings/Depth", "Value:=" , "2.4mm"]
	        ]
	      ]
	   ])
	``

ClearMessages

Clears information in the **Messages** window.

> **Note:**
>
> Additional options are available when using the Desktop-level ClearMessages command.

UI Access	N/A
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ClearMessages()
Python Example	`oProject.ClearMessages()`

CloneMaterial

Clones a local material.

UI Access	N/A
Parameters	Name | Type | Description *<matName>* | String | Name of existing material. *<newName>* | String | Name for newly cloned material.
Return Value	Boolean: • 1 - Material is cloned. • 0 - Existing material not found or a conflict with the new material name.

Python Syntax	CloneMaterial (*<matName>*, *<newName>*)
Python Example	`oDefinitionManager.CloneMaterial("copper1", "copper3")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Close

Closes the active project.

> **Warning:**
>
> Unsaved changes will be lost.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	Close()
Python Example	`oProject.Close()`

CopyDesign

Copies a specified design.

UI Access	**Edit** > **Copy**.
Parameters	Name | Type | Description *<DesignName>* | String | Name of the design to copy from.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CopyDesign (<*DesignName*>)
Python Example	`oProject.CopyDesign ("HFSSDesign1")`

CutDesign

Cuts a design from the active project. The design is stored in memory and can be pasted.

> **Warning:**
>
> This is a legacy command that is no longer supported and should not be used as it may have unintended effects on solved designs.

UI Access	**Edit** > **Cut**.
Parameters	Name | Type | Description <*DesignName*> | String | Name of the design.
Return Value	None.

Python Syntax	CutDesign (<*DesignName*>)
Python Example	`oProject.CutDesign("SimplorerDesign1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteDataset

Deletes a specified dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Remove**.
Parameters	Name | Type | Description <*DatasetName*> | String | Name of the dataset found in the project.
Return Value	None.

Python Syntax	DeleteDataset (<*DatasetName*>)
Python Example	`oProject.DeleteDataset('$ds1')` `oDesign.DeleteDataset('$ds1')`

DeleteDesign

Deletes a specified design in the project.

UI Access	**Edit** > **Delete**, or **Delete** in the ribbon.
Parameters	Name | Type | Description <*DesignName*> | String | Name of the design.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteDesign (<*DesignName*>)
Python Example	`oProject.DeleteDesign("HFSS 3D LayoutDesign2")`

DeleteToolObject

> **Note:**
> This command is for internal Ansys use only.

UI Access	N/A
Parameters	Name | Type | Description <*ObjectName*> | String | Name of the tool object.
Return Value	None.

Python Syntax	DeleteToolObject(<*ObjectName*>)
Python Example	`oProject.DeleteToolObject("object1")`

EditDataset

Modifies a dataset. This can be executed by the oProject, or oDesign variables.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Project > Datasets > Edit.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><OriginalName></td><td>String</td><td>Name of the original dataset.</td></tr>
<tr><td><DatasetDataArray></td><td>Array</td><td>Data for the modified dataset.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	EditDataset (<*OriginalName*> <*DatasetDataArray*>)

Python Example

```
oProject.EditDataset ("ds1"
["NAME:ds2",
  ["NAME:Coordinates",
    [
      "NAME:Coordinate",
      "X:=", 1, "Y:=", 2
    ],
    [
      "NAME:Coordinate",
      "X:=", 3, "Y:=", 4
    ]
  ]
]
)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oDesign.EditDataset ("ds1"
["NAME:ds2",
  ["NAME:Coordinates",
    [
      "NAME:Coordinate",
      "X:=", 1, "Y:=", 2
    ],
    [
      "NAME:Coordinate",
      "X:=", 3, "Y:=", 4
    ]
  ]
]
)
```

EditMaterial

Modifies an existing material.

UI Access	**View/Edit Materials** command in the material editor
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<OriginalName>	String	Name of the material before editing.

<MatProperties>	Array	Structured array containing material properties:
		`[`
		`"NAME:<New material name>",`
		`"CoordinateSystemType:=", <string>,`
		`"BulkOrSurfaceType:=" , <integer>,`
		`[`
		`"NAME:PhysicsTypes",`
		`"set:=" , <array containing string physics types>`
		`],`
		`<Optional ModifierDataArray>,`
		`"permeability:=" , <string containing value>,`
		`"conductivity:=" , <string containing value>,`
		`"thermal_conductivity:=", <string containing value>,`
		`"mass_density:=" , <string containing value>,`
		`"specific_heat:=" , <string containing value>,`
		`"youngs_modulus:=" , <string containing value>,`
		`"poissons_ratio:=" , <string containing value>,`
		`"thermal_expansion_coefficient:=", <string containing value>`
		`]`

<Modi-fierDataArray>	Array	Optional structured array containing thermal or spatial modifiers:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  [
  "NAME:ModifierData",
  [
    "NAME:<ThermalModifierData or SpatialModifierData>",
    "modifier_data:=" , <"thermal_modifier_data" or "spa-
    tial_modifier_data">,
    [
      "NAME:<all_thermal_modifiers or all_spatial_mod-
      ifiers>",
      [
        "NAME:<modifierName>",
        "Property::=" , <string property being mod-
        ified>,
        "Index::=" , <integer>,
        "prop_modifier:=" , <"thermal_modifier" or "spa-
        tial_modifier">,
        "use_free_form:=" , <Boolean>,
        "free_form_value:=" , <string modifier value>,
      ]
    ]
  ]
]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	EditMaterial (<*OriginalName*>, <*MatProperties*>)

Python Example	**Without Modifiers:**
	`oDefinitionManager.EditMaterial("alumina_92pct",`
	`[`
	`"NAME:alumina_92pct",`
	`"CoordinateSystemType:=" , "Cartesian",`
	`"BulkOrSurfaceType:=" , 1,`
	`[`
	`"NAME:PhysicsTypes",`
	`"set:=" , ["Electromagnetic","Thermal","Structural"]`
	`],`
	`"permittivity:=" , "9.3",`
	`"dielectric_loss_tangent:=" , "0.008",`
	`"thermal_conductivity:=" , "26",`
	`"mass_density:=" , "3720",`
	`"specific_heat:=" , "790",`
	`"youngs_modulus:=" , "267000000000",`
	`"poissons_ratio:=" , "0.26",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "thermal_expansion_coeffcient:=", "7.2e-006"
]
)
```

With Thermal Modifier:

```
oDefinitionManager.EditMaterial("copper",
   [
   "NAME:copper",
   "CoordinateSystemType:=", "Cartesian",
   "BulkOrSurfaceType:=" , 1,
      [
      "NAME:PhysicsTypes",
      "set:=" , ["Electromagnetic","Thermal","Structural"]
      ],
   [
   "NAME:ModifierData",
      [
      "NAME:ThermalModifierData",
      "modifier_data:=" , "thermal_modifier_data",
         [
         "NAME:all_thermal_modifiers",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          [
          "NAME:one_thermal_modifier",
          "Property:=" , "permittivity",
          "Index:=" , 0,
          "prop_modifier:=" , "thermal_modifier",
          "use_free_form:=" , True,
          "free_form_value:=" , "if(Temp > 1000cel, 1, if(Temp < -273.15cel, 1, 1))"
          ]
       ]
     ]
  ],
  "permeability:=" , "0.999991",
  "conductivity:=" , "58000000",
  "thermal_conductivity:=" , "400",
  "mass_density:=" , "8933",
  "specific_heat:=" , "385",
  "youngs_modulus:=" , "120000000000",
  "poissons_ratio:=" , "0.38",
  "thermal_expansion_coefficient:=" , "1.77e-05"
])
```

Transient Solve, Non-linear Drude Data Plasma

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Drude_plasma_parameters_r231")
oDefinitionManager = oProject.GetDefinitionManager()
oDefinitionManager.EditMaterial("Drude",
  [
    "NAME:Drude",
    "CoordinateSystemType:=", "Cartesian",
    "BulkOrSurfaceType:="   , 1,
    [
      "NAME:PhysicsTypes",
      "set:="                , ["Electromagnetic"]
    ],
    [
      "NAME:AttachedData",
      [
        "NAME:MatNonLinearDrudeFreqDepData",
        "property_data:="     , "nonlinear_drude_data",
        "EpsilonInfinity:="    , "1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "PlasmaFrequency:="    , "4.62348462366278GHz",
            "CollisionFrequency:="  , "0.00054491190162662GHz",
            "FieldBreakdown:="     , "10000V_per_meter",
            "PlasmaMaintainFrequency:=", "2.31174231183139GHz",
            "NeutralDensity:="     , 2.65164580488373E+20,
            "ElectronDensity:="    , 2.65164580488373E+17,
            "CollisionRateConstant:=", 2.05499505485618E-15
        ]
    ]
])
```

ExportDataset

Exports a dataset to a named file. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Export**.
Parameters	Name | Type | Description *<datasetFileFullPath>* | String | The full path to the file.
Return Value	None.

Python Syntax	ExportDataset (<*datasetFileFullPath*>)
Python Example	`oProject.ExportDataset('e:/tmp/dsdata.txt')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oDesign.ExportDataset('e:/tmp/dsdata.txt')`

ExportMaterial

Exports a local material to a library.

UI Access	**Export to Library** command in the material editor.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ExportData*></td><td>Array</td><td>["NAME:<LibraryName>", <MaterialName>, <MaterialName>, ...]</td></tr><tr><td><*LibraryName*></td><td>String</td><td>Name of the exported library.</td></tr><tr><td><*MaterialName*></td><td>String</td><td>Name of the material to be exported.</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location to save the library. Only "PersonalLib" and "UserLib" are allowed.</td></tr></table>
Return Value	None.

Python Syntax	ExportMaterial (<*ExportData*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.ExportMaterial (["NAME:mylib",` `"Material1", "Material2", "Material3"], "PersonalLib")`

GetActiveDesign

Returns the design in the active project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note: `GetActiveDesign` will return normally if there are no active objects.

UI Access	N/A
Parameters	None.
Return Value	Object of the active design.

Python Syntax	GetActiveDesign()
Python Example	`oDesign = oProject.GetActiveDesign()`

GetArrayVariables

Returns a list of array variables. To get a list of indexed project variables, execute with oProject. To get a list of indexed local variables, use oDesign.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing names of variables.

Python Syntax	GetArrayVariables()
Python Example	`oProject.GetArrayVariables()` `oDesign.GetArrayVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChildNames [Project]

Returns the names of the project's child objects.

UI Access	N/A
Parameters	Name | Type | Description *<type>* | String | (Optional) Default is "design"; Any value returned by GetChildTypes() can be used.
Return Value	Array of names of children for the queried object.

Python Syntax	GetChildNames (*<type>*)
Python Example	`arrDesignNames = oProject.GetChildNames()` `arrVarbleNames = oProject.GetChildNames("Variable")`

GetChildObject [Project]

Returns the project's child objects.

> **Note:**
>
> `GetChildObject` will return normally if there are no active objects.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*path*></td><td>String</td><td>The path may include multiple generations (for example, "designObject/moduleObj/SetupObject"). See Object Path.</td></tr></table>
Return Value	Object of a found child.

Python Syntax	GetChildObject(<*path*>)

Python Example	`oProject = oDeskop.GetActiveProject()`
	`oDesign = oProject.GetChildObject(“TeeModel”)`
	`oVariable = oProject.GetChildObject("VariableName")`
	`oReport = oProject.GetChildObject("TeeModel/Results/S Parameter Plot 1")`

GetChildTypes [Project]

Returns the types of the project's child objects.

UI Access	N/A
Parameters	None.
Return Value	Array of string represents types of the child objects.

Python Syntax	GetChildTypes()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oProject.GetChildTypes()`

GetDefinitionManager

Gets the `DefinitionManager` object.

UI Access	N/A
Parameters	None.
Return Value	`DefinitionManager` object.

Python Syntax	GetDefinitionManager()
Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`

> **Note:** For more information on commands for the DefinitionManager, see Definition Manager Script Commands.

GetDependentFiles

Provides a list of the external files referenced in the project, including characteristic (for example, MDX) and coupled project files.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	List of referenced files.

Python Syntax	GetDependentFiles()
Python Example	`files = oProject.GetDependentFiles()`

GetDesign

Returns the interface to a specific design in a given project.

UI Access	N/A
Parameters	Name | Type | Description *<designName>* | String | Name of the design.
Return Value	Object of the specified design.

Python Syntax	GetDesign (*<designName>*)
Python Example	`oProject.GetDesign("HFSSDesign1")`

GetDesigns

Obtains all designs in the current project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	List of objects for all designs in the project.

Python Syntax	GetDesigns()
Python Example	`oProject.GetDesigns()`

GetEDBHandle

Returns the EDB handle for the project.

> **Important:**
>
> This script is for internal Ansys use only.

UI Access	N/A
Parameters	None.
Return Value	String indicating the EDB handle for the project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetEDBHandle()
Python Example	`oProject.GetEDBHandle()`

GetLegacyName

Obtains the legacy name of a project.

> **Note:**
> This command is for internal Ansys use only.

UI Access	N/A
Parameters	None.
Return Value	String containing the legacy project name.

Python Syntax	GetLegacyName()
Python Example	`oProject.GetLegacyName()`

GetName [Project]

Obtains the project name.

UI Access	N/A
Parameters	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	String containing the project name, not including the path or extension.

Python Syntax	GetName()
Python Example	`oProject.GetName()`

GetObjPath [Project]

Obtains the project name from the full path.

UI Access	N/A
Parameters	None.
Return Value	String containing only the project name.

Python Syntax	GetObjPath()
Python Example	`oProject.GetObjPath()`

GetPath

Returns the location of the project on disk.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the project, not including the project name.

Python Syntax	GetPath()
Python Example	`oProject.GetPath()`

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	Name | Type | Description <*PropName*> | String | Name of the property.
Return Value	String value of the evaluated value.

Python Syntax	GetPropEvaluatedValue (<*PropName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oVar = oDesign.GetChildObject(" Variables/var")` `oVar.GetPropEvaluatedValue()`

GetPropNames [Project]

Obtains the property name of the object. At the project level, GetPropNames always returns empty because the project is not associated with any property.

UI Access	N/A
Parameters	Name: *<includeReadOnly>* Type: Boolean Description: (Optional) • True – Include read only props. • False – Do not include read only props.
Return Value	Empty array.

Python Syntax	GetPropNames ()
Python Example	`oProject.GetPropNames()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropSIValue

Returns the SI-Value for Value-Property and Variable. Return NAN for other property type if its value is not able to convert to be a double-floating point value.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PropName*></td><td>String</td><td>Name of the property.</td></tr></table>
Return Value	Property value as a double floating value, or NAN if the property value cannot be converted to double floating point.

Python Syntax	GetPropSIValue (<*PropName*>)

Python Example	`oCreateBox = oDesign.GetChildObject("3D Modeler/Box1/CreateBox:1")`
	`oCreateBox.GetPropValue("xSize")`
	return "length / 2"
	`oCreateBox.GetPropEvaluatedValue("xSize")`
	return '0.4mm'
	`oCreateBox.GetPropSIValue("xSize")`
	return 0.0004

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropValue [Project]

Returns the property value for the active project object, or specified property values.

UI Access	N/A
Parameters	Name: *<propPath>* — Type: String — Description: A child object's property path. See: Property Function Summary.
Return Value	String of property value.

Python Syntax	GetPropValue(*<propPath>*)
Python Example	`oProject.GetPropValue("TeeModel/offset")`

GetProperties

Gets a list of all the properties belonging to a specific <PropServer> and <PropTab>. This can be executed by the oProject, oDesign, or oEditor variables.

UI Access	N/A		
Parameters	Name: *<PropTab>*	Type: String	Description: One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<PropServer>	String	An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3
Return Value	Array of strings containing the names of the appropriate properties.		

Python Syntax	GetProperties(<*PropTab*>, <*PropServer*>)
Python Example	`oEditor.GetProperties('PassedParameterTab', 'k')`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

> **Tip:** Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*PropTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
<*PropServer*>	String	An object identifier, generally returned from another script method, such as `Comp‡nst@R;2;3`
<*PropName*>	String	Name of the property.

Return Value

String value of the property.

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	. . .

GetTopDesignList

Returns a list of top-level design names.

UI Access	N/A
Parameters	None.
Return Value	List of strings containing name of top-level designs.

Python Syntax	GetTopDesignList()
Python Example	`oProject.GetTopDesignList()`

GetVariableValue

Gets the value of a single specified variable. To get the value of project variables, execute this command using `oProject`. To get the value of local variables, use `oDesign`.

UI Access	N/A
Parameters	Name | Type | Description <*VarName*> | String | Name of the variable to access.
Return Value	String represents the value of the variable.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetVariableValue(<*VarName*>)
Python Example	`oProject.GetVariableValue("var_name")`

GetVariables

Returns a list of all defined variables. To get a list of project variables, execute this command using `oProject`. To get a list of local variables, use `oDesign`.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the variables.

Python Syntax	GetVariables ()
Python Example	`oProject.GetVariables()` `oDesign.GetVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportDataset

Imports a dataset from a named file. This can be executed by the oProject, or oDesign variables. The name of the dataset is file-name+index number (e.g., dsdata1) unless the filename ends with a trailing number. When there is a trailing number at the end, we will remove the number and use first unused index. Alternatively, the name of the dataset can be explicitly defined by providing a string as an optional second argument.

UI Access	**Project** > **Datasets** > **Import**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*datasetFileFullPath*></td><td>String</td><td>The full path to the file containing the dataset values. *.tab files recommended (see note below).</td></tr><tr><td><*optionalDatasetName*></td><td>String</td><td>*Optional*. User-defined dataset name.</td></tr></table>
Return Value	None.

Python Syntax	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)

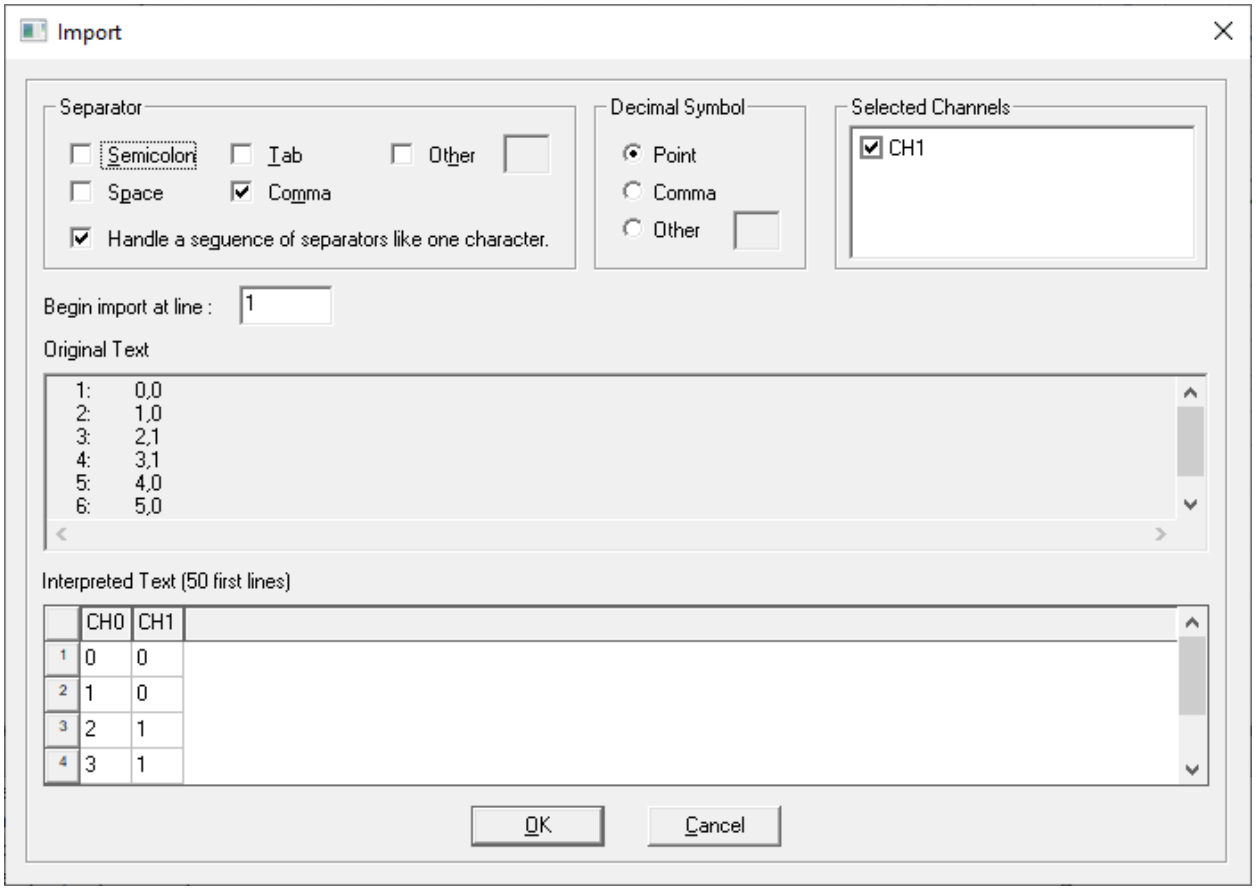
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

Note About File Types:

Tab-delimited or space-delimited files with the extension *.tab are the recommended file type. When using ImportDataset at the Design level, *.tab is the only file type supported.

At the Project level, other file types are supported (for example, *.csv). However, after calling the command, you must configure the file import format manually through the Electronics Desktop GUI by selecting **Project** > **Datasets** and clicking **Import**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



InsertDesign

Inserts a new design in the project. For Circuit and HFSS 3D Layout scripts, the last argument will always be empty.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Project** > **[Insert Circuit Design / Insert Circuit Netlist / Insert HFSS 3D Layout Design]**.
Parameters	

Name	Type	Description
<*DesignType*>	String	Design type. Valid types: "Circuit", "Nexxim Circuit", "System" , "Nexxim Net-list", "HFSS3D".
<*DesignName*>	String	Design name.
<*TechnologyFile*>	String	The path to the Circuit technology file to be used in the design. Use a pair of empty double quotes ("") for none.
<*SubcircuitID*>	String	(Optional) Must be preceded by a colon if included along with the Technology File name. No colon is necessary when the subcircuit ID is omitted.
""	None	Empty argument.

Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*TechnologyFile*>:<*SubCircuitID*>,")
Python Example	`oProject.InsertDesign("Nexxim Circuit","MyDesign", "C:\\Program Files\\ANSYS Inc\\v261\\AnsysEM", "")`

InsertDesign (Layout Editor)

Insert a design

Command: None

Syntax: InsertDesign <type>, <name>, <stationerypath>, <parentname>

Return Value: None

Parameters: <type>

Type: string

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<name>

Type: string

<stationerypath>

Type: string

<parentname>

Type: string

InsertDesignWithWorkflow

Inserts a design with a named workflow and returns an IDispatch string.

UI Access	N/A
Parameters	

Name	Type	Description
<*type*>	String	Type of design.
<*workflowName*>	String	Name of the workflow.
<*specName*>	String	Name of the spec.
<*fileName*>	String	Name of the file.
<*libLoc*>	String	Type of library, such as SysLib.
<*stationaryPath*>	String	Path.

Return Value	IDispatch string, such as 'IDispatch(IAltraSimScript)'

Python Syntax	InsertDesignWithWorkflow(<*type*>, <*workflowName*>, <*specName*>, <*fileName*>, <*libLoc*>, <*stationaryPath*>)
Python	`oProject.InsertDesignWithWorkflow ("Circuit Design", "Serial Design",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example	"PCIe3 Stressed", "LongChannel", "SysLib", "C:\\Program Files\\ANSYS Inc\\v261\\AnsysEM\\syslib\\MS" - RT_duroid 6010 (Er=10.2) 0.010 inch, 0.5 oz copper.asty")

InsertToolObject

> **Note:**
> This command is for internal Ansys use only.

Python Syntax	InsertToolObject()
Python Example	oProject.InsertToolObject()

Paste (Project Object)

Pastes a design in the active project.

UI Access	**Edit** > **Paste**.
Parameters	None.
Return Value	None

Python Syntax	Paste()
Python Example	oProject.Paste()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Redo [Project Level]

Reapplies the last project-level command.

UI Access	**Edit** > **Redo**.
Parameters	None.
Return Value	None.

Python Syntax	Redo()
Python Example	`oProject.Redo()`

RemoveAllUnusedDefinitions

Removes all unused project definitions.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	RemoveAllUnusedDefinitions()
Python Example	`oProject.RemoveAllUnusedDefinitions()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveMaterial

Removes a material from a library.

UI Access	**Remove Material(s)** command in the material editor
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*MaterialName*></td><td>String</td><td>Name of the material to be removed.</td></tr><tr><td><*IsProjectMaterial*></td><td>Boolean</td><td>If True, assumes the material is a project material. The last two parameters will be ignored. If False, the material is not a project material.</td></tr><tr><td><*LibraryName*></td><td>String</td><td>Name of the user or personal library where the material resides.</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location of library. Valid options:"UserLib" or "PersonalLib".</td></tr></table>
Return Value	None.

Python Syntax	RemoveMaterial (<*MaterialName*>, <*IsProjectMaterial*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveMaterial ([` `"Material1", false, "mo0907","UserLib"])`

RemoveUnusedDefinitions

Removes any unused project definitions.

UI Access	**Tools** > **Project Tools** > **Remove Unused Definitions**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Definitions*></td><td>Array</td><td>Definitions to be removed, such as materials and surface materials.</td></tr></table>
Return Value	None.

Python Syntax	RemoveUnusedDefinitions(<*Definitions*>)

Python Example	`oProject.RemoveUnusedDefinitions(`
	`  [`
	`  [`
	`  "NAME:Materials",`
	`  "Al-Extruded"`
	`  ],`
	`  [`
	`  "NAME:SurfaceMaterials",`
	`  "Steel-oxidised-surface"`
	`  ]`
	`  ])`

Rename

Renames the project and saves it. Similar to SaveAs().

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Edit** > **Rename**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*NewName*></td><td>String</td><td>Desired name of the project. The path is optional.</td></tr><tr><td><*OverWriteOk*></td><td>Boolean</td><td>• True - overwrite the file on disk if it exists. • False - prevent overwrite.</td></tr></table>
Return Value	None.

Python Syntax	Rename(<*NewName*>,<*OverWriteOK*>)
Python Example	`oProject.Rename("c:\projects\MyProject.aedt", True)`

RunToolkit

Runs a Python toolkit script, applying it to the active design. The toolkit script itself may have prerequisites, such as sweeps defined, or specific model characteristics, such as port definitions.

Important:

Full scripting for cable modeling is not supported and no arguments are allowed in the script. The **RunToolkit** command will not automatically create a cable bundle, but will simply open the **Cable Modeling** dialog box. You will have to manually input your parameters.

UI Access	[**HFSS 3D Layout**] > **Toolkit** > [**Script Name**].
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*LibraryType*></td><td>String</td><td>Name of the library in which the script is located.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*ToolkitName*>	String	Name of the Python script.
	<*ToolkitArg*>	Array	Structured array containing arguments required by the toolkit script.
Return Value	User-defined solution(s) and/or output(s).		

Python Syntax	RunToolkit(<*LibraryType*>, <*ToolkitName*>, <*ToolkitArg*>)
Python Example	`oDesign.RunToolkit("SysLib", "Hearing Aid Compliance", [])`

Save

Saves the active project.

UI Access	**File** > **Save**.
Parameters	None.
Return Value	None.

Python Syntax	Save()
Python Example	`oProject.Save()`

SaveAs

Saves the project under a new name. Requires a full path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

This script takes two parameters for non-schematic/layout designs and four parameters for schematic/layout designs.

<table>
<tr><td>UI Access</td><td>File > Save As.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><NewName></td><td>String</td><td>The desired name of the project, with directory and extension.</td></tr>
<tr><td><OverWriteOK></td><td>Boolean</td><td>True to overwrite the file of the same name, if it exists. False to prevent overwrite.</td></tr>
<tr><td><DefaultAction></td><td>String</td><td>For Schematic/Layout projects only. Otherwise omit. See note below.
Valid actions: ef_overwrite , ef_copy_no_overwrite, ef_make_path_absolute, or empty string.</td></tr>
<tr><td><OverwriteActions></td><td>Array</td><td>For Schematic/Layout projects only. Otherwise omit. See note below.
Structured array: ["Name: <Action>", <FileName>, <FileName>, ...]
Valid actions: ef_overwrite , ef_copy_no_overwrite, ef_make_path_absolute, or empty string.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>For non-Schematic/Layout project: SaveAs (<NewName> <OverWriteOK>)
For Schematic/Layout project: SaveAs (<NewName> <OverWriteOK> <DefaultAction> <OverrideActions>)</td></tr>
<tr><td>Python Example</td><td><pre>oProject.SaveAs('D:/projects/project1.aedt', True)</pre>----<pre>oProject.SaveAs('D:/Projects/Project1.aedt', True, 'ef_overwrite', ['NAME:OverrideActions', ['NAME:ef_copy_no_overwrite', ['NAME:Files',</pre></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`'$PROJECTDIR/circuit_models.inc']], ['NAME:ef_make_path_absolute', ['NAME:Files', '$PROJECTDIR/SL_6s.sp']]])`

Important:

The DefaultAction and OverrideActions strings correspond to the following actions:

- **ef_overwrite** – Copy file to new project directory and overwrite.
- **ef_copy_no_overwrite** – Copy file to new project directory and don't overwrite.
- **ef_make_path_absolute** – Change reference to point to file in old project directory.
- **Empty String** – Do nothing.

The DefaultAction is applied to all files that are NOT explicitly listed in the OverrideActions array. Those in the OverrideActions array are separate arrays for actions that are different from the default action; those actions are applied to the files listed in the same array:

- If OverrideActions are not specified, DefaultAction is applied to ALL files in project directory.

SaveAsStandAloneProject

Saves the project as a standalone copy.

Note:

This script is not supported when the application is being controlled by Ansys Workbench.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description <*projectName*> | String | The desired name of the project, with directory and extension.
Return Value	None.

Python Syntax	SaveAsStandAloneProject(<*projectName*>)
Python Example	`oProject.SaveAsStandAloneProject('D:/projects/project1.aedt')`

SaveProjectArchive

Saves the active project as an archive to the specified file path.

UI Access	**File** > **Archive**.
Parameters	Name | Type | Description <*archiveFilePath*> | String | Path to archived file. <*IncludeExternalFiles*> | Boolean | True to include external files; False to exclude. <*IncludeResultsFiles*> | Boolean | True to include simulation files associated with the project; False to exclude. <*AdditionalFiles*> | Array | Additional specified files to include. <*ArchiveNotes*> | String | String describe the archive.
Return Value	None.

Python Syntax	SaveProjectArchive(<*archivefilepath*>, <*IncludeExternalFiles*>, <*IncludeResultsFiles*>, <*AdditionalFiles*>, <*ArchiveNotes*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oProject.SaveProjectArchive("C:\\Users\\Documents\\Ansoft\\Project27.aedtz", True, False, [], "")`

SetActiveDefinitionEditor

Obtains a specified definition editor.

UI Access	N/A		
Parameters	Name	Type	Description
	<EditorName>	String	Name of the definition editor to set active, one of "SymbolEditor", "FootprintEditor".
	<DefinitionName>	String	The combination name for the symbol or footprint, *<libname>*:*<def-name>*
Return Value	Object for the definition to be edited.		

Python Syntax	SetActiveDefinitionEditor(*<EditorName>*, *<DefinitionName>*)
Python Example	`oProject.SetActiveDefinitionEditor("SymbolEditor",` `"Simplorer Elements\Basic Elements\Circuit\Passive Elements:R")`

SetActiveDesign

Sets a design to be the active design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name | Type | Description <*DesignName*> | String | Name of the design to set as the active design.
Return Value	None.

Python Syntax	SetActiveDesign (<*DesignName*>)
Python Example	`oDesign = oProject.SetActiveDesign("SimplorerDesign2")`

SetPropValue [Project]

Sets a property value for an active project's child object.

UI Access	Edit Properties on ProjectTree objects.
Parameters	Name | Type | Description <*propPath*> | String | A child object's property path. See: Property Function. <*newValue*> | String | New property value.
Return Value	Boolean: • **True** – property found. • **False** – property not found.

Python Syntax	SetPropValue(<*propPath*>, <*newValue*>)
Python Example	`oProject.SetPropValue("TeeModel/offset", "2mm")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oProject.SetPropValue("TeeModel/Results/S Parameter Plot 1/Display Type", "Data Table")`

SetPropertyValue

Sets the value of a single property belonging to a specific PropServer and PropTab. This function is available with the Project, Design or Editor objects, including definition editors.This is not supported for properties of the following types: ButtonProp, PointProp, V3DPointProp, and VPointProp. Only the ChangeProperty command can be used to modify these properties.

Use the script recording feature and edit a property, and then view the resulting script entry or use GetPropertyValue for the desired property to see the expected format.

<table>
<tr><td>UI Access</td><td colspan="3">N/A</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><propTab></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:
PassedParameterTab ("Parameter Values")
DefinitionParameterTab (Parameter Defaults")
LocalVariableTab ("Variables" or "Local Variables")
ProjectVariableTab ("Project variables")
ConstantsTab ("Constants")
BaseElementTab ("Symbol" or "Footprint")
ComponentTab ("General")
Component("Component")
CustomTab ("Intrinsic Variables")
Quantities ("Quantities")
Signals ("Signals")</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*propServer*>	String	An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3
	<*propName*>	String	Name of the property.
	<*propValue*>	String	The value for the property
Return Value	None.		

Python Syntax	SetPropertyValue(<*propTab*>, <*propServer*>, <*propName*>, <*propValue*>)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

SetVariableValue

Sets the value of a variable. To set the value of a project variable, execute this command using `oProject`. To set the value of a local variable, use `oDesign`.

UI Access	N/A		
Parameters	Name	Type	Description
	<*VarName*>	String	Variable name.
	<*VarValue*>	Value	New value for the variable.
Return Value	None.		

Python Syntax	SetVariableValue (<*VarName*>, <*VarValue*>)
Python Example	`oProject.SetVariableValue('$Var1', '3mm')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SimulateAll

Simulates all solution setups and Optimetrics setups for all design instances in the project. Script processing only continues when all analyses are finished.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	SimulateAll()
Python Example	`oProject.SimulateAll()`

Undo [Project]

Cancels the last project-level command.

UI Access	**Edit** > **Undo**.
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Undo()
Python Example	`oProject.Undo()`

UpdateDefinitions

Updates all definitions. The **Messages** window reports when definitions are updated, or warns when definitions cannot be found.

UI Access	**Tools** > **Project Tools** > **Update Definitions**. Click **Select All**, then **Update**.
Parameters	None.
Return Value	None.

Python Syntax	UpdateDefinitions()
Python Example	`oProject.UpdateDefinitions()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

7 - Design Object Script Commands

Design object commands should be executed by the oDesign object.

```
oDesign.CommandName <args>
```

For example:

```
oDesign = oProject.SetActiveDesign("HFSSDesign1")
```

Conventions Used in this Chapter

`<ModuleName>` is a placeholder for any one of the following modules:

- Excitations Commands – "Excitations" or "Boundaries"
- Field Overlays Module – "FieldsReporter"
- Optimetrics Module – "Optimetrics"
- Output Variable Module – "OutputVariable"
- Radiation Setup Manager Module – "RadiationSetupMgr"
- Reporter Module – "ReportSetup"
- Simulation Setup Commands – "SolveSetups"
- User Defined Documents Module – "UserDefinedDocuments"
- User Defined Solutions Module – "UserDefinedSolutionModule"

AddDataset

AddDynamicLink

AddModelingProperties

Analyze

AnalyzeAll

AnalyzeAllNominal

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ApplyMeshOps

ChangePortProperty [Nexxim]

ChangeProperty

CopyItemCommand

CreateReport

DeleteDataset

DeleteDesignInstance

DeleteFieldVariation

DeleteFullVariation

DeleteOutputVariable

DeletePlotEntities

DeletePort [Nexxim]

DeleteSource [Nexxim]

EditCoSimulationOptions

EditDataset

EditDesignSettings

EditImportData

EditInfiniteArray

EditLayoutForLayoutComponent

EditNotes

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditNotes (Layout Editor)

EditOptions

EditOutputVariable

EMDesignOptions

ExportBroadbandRLGCNetlist

ExportCircuit (Layout Editor)

ExportConvergence

ExportDataset

ExportForHSpice

ExportForHSpice (Layout Editor)

ExportForSpice (Layout Editor)

ExportNMFData

ExportLayoutViaCurrentDensity

ExportMeshStats

ExportNetworkData

ExportNetworkData (Layout Editor)

ExportNMFData

ExportProfile

ExportReport

GenerateMesh

GetAllPorts

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChildNames [Design]

GetChildObject [Design]

GetChildTypes [Design]

GetDesignID

GetDesignType

GetDesignName

GetDesignValidationInfo

GetEditor

GetEditor (Layout Editor)

GetGeometryIdsForAllNetLayerCombinations

GetGeometryIdsForNetLayerCombinations

GetManagedFilesPath

GetModule

GetName

GetNominalVariation

GetNoteText

GetObjPath

GetOutputVariableValue

GetOutputVariables

GetProject

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPostProcessingVariables

GetPropHost

GetPropNames [Design]

GetPropValue [Design]

GetProperties

GetPropertyValue

GetSelections

GetSolutionType

GetSolveInsideThreshold

GetVariableValue

GetVariables

ImportDataFilePath [Nexxim]

ImportDataset

InsertDesign

OverlayCurrents

OverlayFarField

OverlayMesh

OverlayNearField

PasteArray

PasteDesign

PasteItemCommand

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PushExcitations

Redo

RemoveImportData (Layout Editor)

RemoveModelingProperties

RenameDesignInstance

RenameImportData [Nexxim]

RenameSource

ReportTemplates

RunToolkit

SetActiveEditor

SetBendGraphicsMode

SetDesignMode

SetPropValue [Design]

SetPropertyValue

SetShowLayoutForLayoutComponent

SetSolveInsideThreshold

SetSourceContexts

SetVariableValue

SimulateLink

Undo

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddDataset

Adds a dataset. This can be executed by the oProject, or oDesign variables.

<table>
<tr><td>UI Access</td><td colspan="3">Project > Datasets > Add</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><*DatasetDataArray*></td><td>Array</td><td>["NAME:<DatasetName>",
["NAME:Coordinates", <CoordinateArray>,
<CoordinateArray>, ...]</td></tr>
<tr><td><*DatasetName*></td><td>String</td><td>Name of the dataset.</td></tr>
<tr><td><*CoordinateArray*></td><td>Array</td><td>["NAME:Coordinate", "X:=", <double>, "Y:=",<double>]</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
</table>

Python Syntax	AddDataset <*DatasetDataArray*>

Python Example	`oProject.AddDataset(`
	`["NAME:$ds1",`
	`  ["NAME:Coordinates",`
	`  ["NAME:Coordinate",`
	`  "X:=", 2,`
	`  "Y:=", 4`
	`  ],`
	`  ["NAME:Coordinate",`
	`  "X:=", 6,`
	`  "Y:=", 8`
	`  ]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``
	  ]
	]
	)
	oDesign.AddDataset(
	[
	"NAME:$ds1",
	  ["NAME:Coordinates",
	  ["NAME:Coordinate",
	  "X:=", 2,
	  "Y:=", 4
	  ],
	  ["NAME:Coordinate",
	  "X:=", 6,
	  "Y:=", 8
	  ]
	  ]
	]
	)
	``

AddDynamicLink

Adds a dynamic link component to a Circuit or HFSS 3D Layout design. Circuit designs can accept dynamic links from HFSS, Q3D, 2D Extractor, and SIwave.

> **Note:**
>
> HFSS 3D Layout designs can only accept dynamic links from HFSS.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>

Open the Circuit or HFSS 3D Layout design to which you want to add the link. On the Component Libraries menu, open the Symbols tab, and then click Import Models. Select the appropriate icon from the panel of icons. The Import Components dialog opens.
If the source for a dynamic link is an HFSS design, in the Project Manager drag and drop the design onto a Circuit or HFSS 3D Layout design to create the link.

</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><designName></td><td>String</td><td>Design source for the dynamic link. Optional, defaults to the first design in the project pointed to by projectPath.</td></tr>
<tr><td><projectPath></td><td>String</td><td>Project source for the dynamic link. Optional, defaults to the current active project.</td></tr>
<tr><td><linkName></td><td>String</td><td>Name applied to the dynamic link. Optional, defaults to the designName.</td></tr>
<tr><td><solutionName></td><td>String</td><td>Solution in source design which will be referenced by the dynamic link. Optional, defaults to the first setup:sweep.</td></tr>
<tr><td><transLinePort ></td><td>String</td><td>For HFSS designs only, port name for transmission lines. It is created only if it is not empty. Optional, defaults to “”. This value is ignored for non-HFSS design types.</td></tr>
<tr><td><ReduceMatrix></td><td>String</td><td>For Q3D and 2D extractor designs, reduce matrix which will be referenced by the dynamic link. This is ignored for other design types. Optional, defaults to “Original”.</td></tr>
<tr><td><EnableHFSSCableModeling></td><td>Boolean</td><td>For 2D extractor designs only, flag to create cable models. This is ignored for other designs types. Optional, defaults to false.</td></tr>
<tr><td><Description></td><td>String</td><td>Description of the dynamic link's source. Optional.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	AddDynamicLink (<*designName*>, <*projectPath*>, <*linkName*>, <*solutionName*>, <*transLinePort*>, <*reducedMatrix*>, <*enableCableModeling*>, <*description*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesign = oProject.SetActiveDesign("Circuit1")` `oDesign.AddDynamicLink("TeeModel","$PersonalLib\\TeeWrongFile.aedt","TeeLink","Setup1 : LastAdaptive","Port1","",False,"desc")`

AddModelingProperties

Use: Add a modeling property to a design

Command: None

Syntax: AddModelingProperties <design>

Return Value: None

Parameters: <design>

Type: string

Analyze

Solves a single solution setup and all of its frequency sweeps.

UI Access	Right-click a solution setup in the project tree, and select **Analyze**.
Parameters	

Name	Type	Description
<setup>	String	Setup name.
isBlocking	Boolean	An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.

Return Value	Integer value: • **0** – Success

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• **Else** – Error

Python Syntax	Analyze (<*setup*>)
Python Example	`oDesign.Analyze("Setup1", isBlocking)`

AnalyzeAll [design]

Runs all solution setups and Optimetrics setups for the current design instance.

UI Access	N/A
Parameters	Name: isBlocking; Type: Boolean; Description: An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	AnalyzeAll()
Python Example	`oDesign.AnalyzeAll(isBlocking)`

AnalyzeAllNominal

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Runs all defined setups.

UI Access	Right-click **Analysis** in the project tree, and select **Analyze All**.
Parameters	Name / Type / Description isBlocking / Boolean / An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	AnalyzeAllNominal()
Python Example	`oDesign.AnalyzeAllNominal()`

ApplyMeshOps

This script has been replaced by GenerateMesh.

ChangeProperty[ReportSetup]

Change the properties for a Report.

UI Access	Double-click on a report to change its properties.
Parameters	Name / Type / Description *<PropertyArray>* / Array / Varies, depending on the properties associated with the select object command.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ChangeProperty(<*PropertyArray*>)

Python Example	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(
	  [
	  "NAME:AllTabs",
	  [
	  "NAME:General",
	  [
	  "NAME:PropServers",
	  "XY Plot 1:General"
	  ],
	  [
	  "NAME:ChangedProps",
	  [
	  "NAME:Use Scientific Notation",
	  "Value:=", True
	  ]
	  ]
	  ]
	])
	
	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(
	  [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NAME:AllTabs",
    [
      "NAME:Legend",
      [
      "NAME:PropServers",
      "S Parameter Plot 1:Legend"
      ],
      [
      "NAME:ChangedProps",
      [
      "NAME:Legend Name",
      "Value:=" , "Ansys"
      ]
    ]
  ]
])
    oModule.ChangeProperty(
    [
      "NAME:AllTabs",
      [
        "NAME:General",
        [
          "NAME:PropServers",
          "S Parameter Plot 1:General"
        ],
    [
      "NAME:ChangedProps",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      [
        "NAME:Auto Scale Fonts",
        "Value:=" , False
      ]
    ]
  ]
])
```

Example For Rectangular Stacked Plot

```
oProject = oDesktop.SetActiveProject("antenna_test")
oDesign = oProject.SetActiveDesign("Design1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Scaling",
      [
        "NAME:PropServers",
        "S Parameter Plot 1:AxisY2:Stack1",
        "S Parameter Plot 1:AxisY2:Stack2"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Min",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "Value:="               , "-60"
          ]
        ]
      ]
    ])
```

Example Change eye rect and eye plot:

```
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("test_nport_synth2")
oDesign = oProject.SetActiveDesign("Circuit1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Grid",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:Grid:Eye"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Major grid line style",
          "Value:=" , "Dot"
        ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:Minor grid line style",
          "Value:=" , "Dot"
        ]
      ]
    ],
    [
      "NAME:Axis",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:AxisX:Eye"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Axis Color",
          "R:=" , 255,
          "G:=" , 128,
          "B:=" , 128
        ]
      ]
    ],
    [
      "NAME:Axis",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:AxisY1:Rect"
      ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:ChangedProps",
          [
            "NAME:Axis Color",
            "R:=" , 255,
            "G:=" , 128,
            "B:=" , 128
          ]
        ]
      ]
])
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Grid",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:Grid:Rect"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Major grid line style",
          "Value:=" , "LongDash"
        ],
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "NAME:Minor grid line style",
            "Value:=" , "LongDash"
          ]
        ]
      ]
    ])
```

CopyItemCommand

Use: Copy tree items, such as Altrasim Solution Setups in Nexxim, or Solve Setups and Frequency Sweeps in Ensemble.

Command: None

Syntax: CopyItemCommand <ItemPathList>

Return Value: None

Parameters: <ItemPathList>

Type: Array of strings

CreateReport

Creates a new report with a single trace and adds it to the **Results** branch in the project tree.

UI Access	Right-click on **Results** > **Create [Type] Report**		
Parameters	Name	Type	Description
	<ReportName>	String	Name of report.
	<ReportType>	String	Type of report. Possible values are: "Modal S Parameters" - Only for Driven Modal solution-type problems

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			with ports. "Terminal S Parameters" - Only for Driven Terminal solution-type problems with ports. "Eigenmode Parameters" - Only for Eigenmode solution-type problems. "Fields" "Far Fields" - Only for problems with radiation or PML boundaries. "Near Fields" - Only for problems with radiation or PML boundaries. "Emission Test"

	<*DisplayType*>	String	Type of display.
			If ReportType is "Modal S Parameters", "Terminal S Parameters", or "Eigenmode Parameters", then set to one of the following:
			`"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Smith Chart", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot", "Rectangular Stacked Plot" or "3D Polar Plot".`
			If <ReportType> is "Fields", then set to one of the following:
			`"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "Rectangular Stacked Plot" or "3D Rectangular Bar Plot".`
			If <ReportType> is "Far Fields" or "Near Fields", then set to one of the following:
			`"Rectangular Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot" "Rectangular`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Stacked Plot" or "3D Polar Plot" If <ReportType> is "Emission Test", then set to one of the following: "Rectangular Plot" or "Data Table"
	<*SolutionName*>	String	Name of the solution as listed in the **Modify Report** dialog box.
	<*ContextArray*>	Array	Context for which the expression is being evaluated. This can be an empty string if there is no context. ["Domain:=", <DomainType>] <DomainType> ex. "Sweep" or "Time" ["Context:=", <GeometryType>] <GeometryType> ex. "Infinite Sphere*n*", "Sphere*n*", "Polyline*n*"
	<*FamiliesArray*>	Array	Contains sweep definitions for the report. ["<VariableName>:= ", <ValueArray>] <ValueArray> ["All"] or ["Value1", "Value2", ..."Value*n*"] examples of <VariableName> "Freq", "Theta", "Distance"
	<*ReportDataArray*>	Array	This array contains the report quantity and X, Y, and (Z) axis definitions. ["X Component:=", <VariableName>, "Y Component:=", <VariableName> | <ReportQuantityArray>] <ReportQuantityArray> ex. ["dB(S(Port1, Port1))"]
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateReport(<*ReportName*>, <*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*ContextArray*>, <*FamiliesArray*>, <*ReportDataArray*>, <*ExtendedInfo*>)

Python Example	Several examples are given below: (nominal, Optimetrics, spectral, Partial Discharge, 3D rectangular Bar Plot with Change Bar Properties, and Add a new orientation to a 3D model and 3D report)
	Nominal Example:
	`oModule.CreateReport( "XY Plot 2", "Standard",`
	`"Rectangular Plot", "TR",`
	`[`
	`"NAME:Context",`
	`"SimValueContext:=", [], 1, 0, 2, 0, false,`
	`false, -1, 1, 0, 1, 1, "", 0, 0]`
	`],`
	`[`
	`"Time:=" , ["All"],`
	`"aaa:=" , ["Nominal"]`
	`],`
	`[`
	`"X Component:=", "Time",`
	`"Y Component:=", ["E1.I"]`
	`],`
	`[])`
	Optimetrics Example:
	`oModule.CreateReport ("XY Plot 3", "Standard",`
	`"Rectangular Plot", "TR",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:Context",
                "OptiSetup:=", "ParametricSetup1", "SimValueContext:=",
                [
                        1, 0, 2, 0, false, false, 0, 1,0, 1, 1, "", 0, 0
                ]
        ],
        [
                "Time:=", ["All"], "aaa:=", ["Nominal"]
        ],
        [
                "X Component:="  , "Time",
                "Y Component:="  , ["E1.I"]
        ],
        [])
```

Spectral Example:

```
oModule.CreateReport ("XY Plot 4", "Standard",
        "Rectangular Plot", "TR",
        [
                "NAME:Context",
                "SimValueContext:=",
                [
                        2, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "",
                        0, 0, "CG", false, "0", "KP",false, "0", "MH",
                        false, "100", "TE", false, "40ms", "TH", false,
                        "40", "TS", false, "0ns", "UF", false, "0",
                        "WT", false, "0", "WW", false, "100"
                ]
        ],
        [
                "Spectrum:="   , ["All"], "aaa:=", ["Nominal"]
        ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "X Component:=","Spectrum",
                "Y Component:=", ["mag(E1.I)"]
        ],
        [])
```

Partial Discharge Example

```
oModule.CreateReport("Partial Discharge Plot 2",
        "Partial Discharge", "Rectangular Plot",
        "PartialDischarge1 : PartialDischarge", [],
        [
                "GasPressure:=", ["All"],
                "Freq:="       , ["All"],
                "bend_angle:=" , ["All"]
        ],
        [
                "X Component:=", "GasPressure",
                "Y Component:=", ["Power"]
        ])
```

Example 3D Rectangular Bar Plot with Change Bar properties

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule = oDesign.GetModule("ReportSetup")
oModule.CreateReport("S Parameter Plot 4", "Modal Solution Data",
       "3D Rectangular Bar Plot", "Setup1 : Sweep", [],
       [
              "Freq:="       , ["All"],
              "UU:="         , ["All"],
              "ZZZ:="        , ["Nominal"]
       ],
       [
              "X Component:=", "Freq",
              "Y Component:=", "UU",
              "Z Component:=", ["dB(S(wDipole1_1_p1,wDipole1_1_p1))"]
       ])

oModule.ChangeProperty(
       [
              "NAME:AllTabs",
              [
                     "NAME:Bar",
                     [
                            "NAME:PropServers"
                            "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                     ],
                     [
                            "NAME:ChangedProps",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                [
                                        "NAME:Transparency",
                                        "Value:=", "0.5"
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Filled",
                                        "Value:=", False
                                ]
                        ]
                ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Tabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Outline Color",
                                        "R:=", 0,
                                        "G:=", 0,
                                        "B:=", 160
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example Script to add a new orientation to a 3D model and 3D report

"AddViewOrientation" script command for 3D Report. The first argument must be the 3D report name.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oDesktop.OpenProject("D:/Tee.aedt")
oProject = oDesktop.SetActiveProject("Project")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.AddViewOrientation("Gain Plot 1", "3dreport", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.634171009063721,
      0.12737800180912,
      -0.762628972530365
    ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    [
      "NAME:Up",
      -0.577874004840851,
      0.577228009700775,
      0.57694798707962
    ]
  ],
  [
    "NAME:ProjectionParams",
      [
      "NAME:Scaling",
      1.10227048397064,
      1.10227036476135,
      1.10227036476135
      ],
      [
      "NAME:Translation",
      -0.601842045783997,
      -0.352552056312561,
      -6.46677732467651
      ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ProjectionType:=" , "Orthographic",
    [
      "NAME:ProjectionLimits",
      -2.53312301635742,
      2.53312301635742,
      -1,
      1,
      0.994183540344238,
      10.5401248931885
    ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.AddViewOrientation("3dmodel", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.627739012241364,
      -0.134194001555443,
      -0.766768991947174
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
      "NAME:Up",
      -0.694571018218994,
      0.54128497838974,
      0.473899990320206
    ]
  ],
  [
  "NAME:ProjectionParams",
    [
      "NAME:Scaling",
      0.0173665378242731,
      0.017366535961628,
      0.0173665322363377
    ],
    [
      "NAME:Translation",
      -0.0742612779140472,
      0.195189163088799,
      -5.89852476119995
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    "ProjectionType:=" , "Orthographic",
      [
        "NAME:ProjectionLimits",
        -2.27072739601135,
        2.27072739601135,
        -1,
        1,
        0.424737930297852,
        10.8078212738037
      ]
    ])
oDesktop.CloseProject("Project")
```

DeleteDataset

Deletes a specified dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Remove**.		
Parameters	Name	Type	Description
	<DatasetName>	String	Name of the dataset found in the project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	DeleteDataset (<*DatasetName*>)
Python Example	`oProject.DeleteDataset('$ds1')` `oDesign.DeleteDataset('$ds1')`

DeleteDesignInstance

Use: Delete the design instance

Command: None

Syntax: DeleteDesignInstance <instance_name>

Return Value: None

Parameters: `<instance_name>`

Type: string

Python Syntax	DeleteDesignInstance(<instanceName>)
Python Example	`oProject.DeleteDesignInstance("Circuit1")`

DeleteFieldVariation

Deletes field variations, fields and meshes, or fields and meshes for specified variations.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	[**Solver**] > **Results** > **Clean Up Solutions** > [**Fields Only** / **Fields and Meshes**].
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*variationKeys*></td><td>Array</td><td>Array containing either "All" string to select all variations, or strings of variations to include.</td></tr><tr><td><*deleteMesh*></td><td>Boolean</td><td>When true, deletes mesh data.</td></tr><tr><td><*deleteLinkedData*></td><td>Boolean</td><td>When true, deletes linked data.</td></tr></table>
Return Value	None.

Python Syntax	DeleteFieldVariation(<*variationKeys*>, <*deleteMesh*>, <*deleteLinkedData*>

Python Example	`oProject = oDesktop.GetActiveProject()`
	`oDesign = oProject.GetActiveDesign()`
	`Design.DeleteFieldVariation(['All'], True, False)`

DeleteFullVariation

Deletes either all solution data, or selected variation data.

UI Access	[**HFSS 3D Layout**] > **Results** > **Clean Up Solutions**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*variationKeys*></td><td>Array</td><td>Array containing either "All" string to select all variations, or strings of variations to include.</td></tr><tr><td><*deleteLinkedData*></td><td>Boolean</td><td>When true, deletes linked data as well.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteFullVariation(<*variationKeys*>, <*deleteLinkedData*>)
Python Example	`oDesign.DeleteFullVariation(['All']), false)`

DeleteOutputVariable

Deletes an existing output variable. The variable can only be deleted if it is not in use by any traces.

UI Access	**HFSS 3D Layout** > **Results** > **Output Variables**. In the **Output Variables** window, click **Delete**.
Parameters	Name | Type | Description <*OutputVarName*> | String | Name of the output variable
Return Value	None.

Python Syntax	DeleteOutputVariable (<*OutputVarName*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.DeleteOutputVariable ("testNew")`

DeleteSolutionVariation

Deletes all solution data for specific solutions and design variations. This is obsolete and is supported only for backward compatibility. You should use DeleteFullVariation.

UI Access	Right-click on **Results**, select **Browse Solutions...**, click **Delete** button in the dialog.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SoluParams*></td><td>Array</td><td>Structured array. `[<DataSpecifierArray>, ...]`</td></tr><tr><td><*DataSpecifierArray*></td><td>Array</td><td>Structured array. `[<DesignVariationKey>, <SetupName>, <SolnName>]`</td></tr></table>
Return Value	None.

Python Syntax	DeleteSolutionVariation(<*SoluParams*>)

Python Example	`oModule.DeleteSolutionVariation([`
	`  ["width='2in'", "Setup1", "Adaptive_1"],`
	`  ["width='2in'", "Setup1", "Sweep1"]`
	`])`

EditCoSimulationOptions

Sets options for cosimulation.

Command: None

Syntax: EditCoSimulationOptions <array_name>

Return Value: None

Parameters: `<array_name>`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

EditDataset

Modifies a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Edit**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*OriginalName*></td><td>String</td><td>Name of the original dataset.</td></tr><tr><td><*DatasetDataArray*></td><td>Array</td><td>Data for the modified dataset.</td></tr></table>
Return Value	None.

Python Syntax	EditDataset (<*OriginalName*> <*DatasetDataArray*>)

Python Example	`oProject.EditDataset ("ds1"`
	`["NAME:ds2",`
	`  ["NAME:Coordinates",`
	`  [`
	`  "NAME:Coordinate",`
	`  "X:=", 1, "Y:=", 2`
	`  ],`
	`  [`
	`  "NAME:Coordinate",`
	`  "X:=", 3, "Y:=", 4`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ]
  ]
]
)
oDesign.EditDataset ("ds1"
["NAME:ds2",
  ["NAME:Coordinates",
    [
      "NAME:Coordinate",
      "X:=", 1, "Y:=", 2
    ],
    [
      "NAME:Coordinate",
      "X:=", 3, "Y:=", 4
    ]
  ]
]
)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditImportData

Use: Edit an imported solution

Command: None

Syntax: EditImportData <oldname> <newlink> <newpath> <newname> <data> <file>

Return Value: None

Parameters: <oldname>

Type: string

<newlink>

<newpath>

Type: string

<newname>

Type: string

<data>

Type: array

<file>

Type: string

EditInfiniteArray

Edits the properties of an infinite array

Command: None

Syntax: EditInfiniteArray <array_name>

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <array_name>

Type: string

Edit (Layout Editor)

Use: Causes modification of one or more existing object(s).

Syntax: **Edit** <Array("NAME:items".....)>,

Parameters: <Array("NAME:items"
<edit_object_info>, // one object info

<edit_object_info>, // another one

…) // etc

<edit_object_info>:

Array("NAME:item",

"name:=", <object_name>, // name of the object

<object_description>) // 'new' object state, type should be consistent with <object_name>:

<object_description>:

<circle_description> |

<rectangle_description> |

<line_description> |

<polyon_description> |

<text_description> |

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<circle_void_description> |
<rectangle_void_description> |
<line_void_description> |
<polyon_void_description> |
<component_description> |
<pin_description> |
<via_description>
```

EditLayoutForLayoutComponent

Opens the definition component for editing in HFSS 3D Layout for a Layout Component in an HFSS 3D design.

UI Access	**Edit Layout...**
Parameters	Name | Type | Description <*ComponentDefinitionName*> | String | Name of the Layout Component Definition.
Return Value	None.

Python Syntax	EditLayoutForLayoutComponent ([Name:LayoutComponent EDB", Definitions:=", [<*ComponentDefinitionName*>])

Python Example	`oDesign.EditLayoutForLayoutComponent (`
	`  [`
	`  "NAME:Layout Component EDB",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Definitions:=" , ["Diff_Via_diffViaNominal"]])

EditNotes

Updates the notes for the design.

UI Access	**HFSS 3D Layout** > **Edit Notes**.		
Parameters	Name	Type	Description
	<DesignNotes>	String	New design notes that are applied, replacing any current notes.
Return Value	None.		

Python Syntax	EditNotes(<*DesignNotes*>)
Python Example	`oDesign.EditNotes("My design notes.")`

EditNotes (Layout Editor)

Edits the notes of a design

Command: None

Syntax: EditNotes <design_name>

Return Value: None

Parameters: <design_name>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

EditOptions (Layout Editor)

Edits the properties of an infinite array

Command: None

Syntax: EditOptions <array_name>

Return Value: None

Parameters: `<array_name>`

Type: string

EditOutputVariable

Changes the name or expression of an existing output variable.

UI Access	N/A

Parameters

Name	Type	Description
<*OrigVarName*>	String	Name of the original output variable
<*NewExpression*>	String	New value to assign to the variable
<*NewVarName*>	String	New name of the variable if any, else pass empty string
<*SolutionName*>	String	Name of the solution as seen in the output variable UI. For example, `"Setup1 : Last Adaptive"`.
<*ReportType*> or <*SolutionType*>	String	Name of the report type as seen in the output variable UI For Layout Editor, use the solution type.
<*ContextArray*> or <*DomainArray*>	Array	Structured array containing context for which the output variable expression is being evaluated. `["Context:=", <string>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			For Layout Editor, use Domain array: `["Domain:=", <string>]`
Return Value	None		

Python Syntax	EditOutputVariable (<*OrigVarName*>, <*NewExpression*>, <*NewVarName*>, <*SolutionName*>, <*ReportType* | *SolutionType*>, <*ContextArray* | *DomainArray*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.EditOutputVariable ("test", "normalize(R1_0.V)", "testNew", "TR", "Standard", [])`

EMDesignOptions

Use: Set options for an EM Design.

Command: Right click on design and select **EM Design Options**

Syntax: DesignOptions <Options Array>

Return Value: None

Parameters:

<Options Array>

Array("NAME:options",

"SaveSolFilesAsBinary:=", <boolean>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"LowPriorityForSimulations:=", <boolean>,

"SaveNearFieldSolutions:=", <boolean>,

"SchematicEnabled:=", <boolean>,

"UseGlobalNumProc:=", <boolean>,

"ComputeBothEvenAndOddCPWModes:=", <boolean>,

"NumProcessors:=", <int>,

"NumProcessorsDistrib:=", <int>,

"CausalMaterials:=", <boolean>,

"UseHPCForMP:=", <boolean>,

"HPCLicenseType:=", <int>)

SaveSolFilesAsBinary - if true, solutions files are saved using a binary format.

LowPriorityForSimulations - if true, run simulations at a lower CPU priority.

SaveNearFieldSolutions - if true, save near field solutions

SchematicEnabled - if true, enable schematics

UseGlobalNumProc - if true, use global number of processors and ignore NumProcessors

ComputeBothEvenAndOddCPWModes - if true, compute both even and odd cpw modes

NumProcessors - number of processors

NumProcessorsDistrib- number of distributed processors

CausalMaterials - if true, use causal materials

UseHPCForMP - if true, use hpc for mp

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HPCLicenseType - number indicating hpc license type

ExportBroadbandRLGCNetlist

Exports Broadband RLGC netlist data.

<table>
<tr><td>UI Access</td><td>From the Project Manager window, right-click a Q3D setup and select Profile > RLGC Matrix Data > Export.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Contents></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:ExportBroadbandRLGCNetlist>
<FileName>
<SolutionName>
<Variation>
<VersionID>
<Pass>
<Frequencies></td></tr>
<tr><td><NAME:ExportBroadbandRLGCNetlist></td><td>String</td><td>Identifies the contents of the array (i.e., "ExportBroadbandRLGCNetlist").</td></tr>
<tr><td><FileName></td><td>String</td><td>Full path to export directory for the new *.cir file.</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution to export data from.</td></tr>
<tr><td><Variation></td><td>String</td><td>Empty string for current nominal variation.</td></tr>
<tr><td><VersionID></td><td>String</td><td>The solution's ID number.</td></tr>
<tr><td><Pass></td><td>String</td><td>The pass to export the data file. If the value is empty or -1, all passes are given.</td></tr>
<tr><td><Frequencies></td><td>Array</td><td>Structured array containing a list of frequencies. If array is empty, all frequencies are used.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportBroadbandRLGCNetlist([<*Contents*>])

Python Example	`oDesign.ExportBroadbandRLGCNetlist(`
	`[`
	`"NAME:ExportBroadbandRLGCNetlist",`
	`"FileName:=" , "C:\\name.cir",`
	`"SolutionName:=" , "Q3D1:Adaptive Pass",`
	`"Variation:=" , "",`
	`"VersionID:=" , 138,`
	`"Pass:=" , -1,`
	`"Frequencies:=" , []`
	`])`

ExportCircuit (Layout Editor)

Exports lumped RLGC data.

UI Access	From the **Project Manager** window, right-click a setup and select **Properties**.		
Parameters	Name	Type	Description
	<Solution>	String	The precise name of the solution from which to export data.
	<SaveLocation>	String	User-selected directory path to save the exported data.
	<CircuitDataArray>	Array	An array that contains these parameters in order: • <NAME:CircuitData>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <MatrixName> • <NumberOfCells> • <UserHasChangedSettings> • <IncludeCap> • <IncludeCond> • <CouplingLimitsArray> • <IncludeDCR> • <IncludeDCL> • <IncludeACR> • <IncludeACL> • <AddResistance> • <IncludeCPP> • <PackageArray> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:CircuitData>	String	Identifies the contents of the array (i.e., CircuitData).
<MatrixName>	String	Original, to export the solved matrix data, or enter the name of any reduced matrices, if available.
<NumberOfCells>	Integer	Number of cells in export.
<UserHasChangedSettings>	Boolean	**True** – export data with the user's chosen settings **False** – export data with the default settings
<IncludeCap>	Boolean	**True** – exports capacitance matrix **False** – does not export capacitance matrix
<IncludeCond>	Boolean	**True** – exports conductance matrix **False** – does not export conductance matrix
<CouplingLimitsArray>	Array	An array that contains these parameters in order:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <NAME:CouplingLimits> • <CouplingLimitType> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:CouplingLimits>	String	Identifies the contents of the array (i.e., CouplingLimits).
<CouplingLimitType>	String	**ByFraction** or **ByValue**
<IncludeDCR>	Boolean	**True** – exports DC resistance matrix **False** – does not export DC resistance matrix
<IncludeDCL>	Boolean	**True** – exports DC inductance matrix **False** – does not export DC inductance matrix
<IncludeACR>	Boolean	**True** – exports AC resistance matrix **False** does not export AC resistance matrix
<IncludeACL>	Boolean	**True** – exports AC inductance matrix **False** – does not export AC inductance matrix
<ADDResistance>	Boolean	**True** – adds DC and AC resistance **False** – does not add the DC and AC resistance **Note:** You cannot export both AC and DC matrices unless **AddResistance** is selected.
<IncludeCPP>	Boolean	**True** – exports data with Chip Package Protocol active **False** – does not export data with Chip Package Protocol active
<RLGCPPInfoArray>	Array	An array that contains these parameters in order: • <NAME:RLGCCPPInfo> • <PackageType> • <LengthUnits> • <NAME:Pins> This array's placeholder does not appear in the syntax or

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			examples below but instead has its contents listed.
	<NAME:RLGCCPPInfo>	String	Identifies the contents of the array (i.e., RLGCCPPInfo).
	<PackageType>	String	**wirebond dieup**, **wirebond diedown**, **flipchip dieup**, or **flip-chip diedown**
	<LengthUnits>	String	A unit of measurement: **um**, **mm**, or **mil**.
	<NAME:Pins>	Array	Identifies the contents of the array (i.e., Pins).
	<Name>	String	The user-entered name of the model.
	<Freq>	Integer	The frequency, in Hz.
Return Value	None		

Python Syntax	oDesign.ExportCircuit(<Solution>, <SaveLocation>, [<CircuitData>, <MatrixName>, <NumberOfCells>, <User-HasChangedSettings>, <IncludeCap>, <IncludeCond>, [<NAME:CouplingLimits>, <CouplingLimitType>], <IncludeDCR>, <IncludeDCL>, <IncludeACR>, <IncludeACL>, <ADDResistance>, <IncludeCPP>, [<NAME:RLGCCPPInfo>, <PackageType>, <LengthUnits>, [<NAME:Pins>]]], <Name>, <Freq>)

Python Example	`oDesign.ExportCircuit("Q3D1 : Last Adaptive", "", "C:/Projects/Q3D/Example.cir",`
	`  [`
	`  "NAME:CircuitData",`
	`  "MatrixName:=" , "Original",`
	`  "NumberOfCells:=" , "1",`
	`  "UserHasChangedSettings:=", True,`
	`  "IncludeCap:=" , False,`
	`  "IncludeCond:=" , False,`
	`  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:CouplingLimits",
      "CouplingLimitType:=" , "None"
    ],
    "IncludeDCR:=" , True,
    "IncludeDCL:=" , True,
    "IncludeACR:=" , False,
    "IncludeACL:=" , False,
    "ADDResistance:=" , False,
    "IncludeCPP:=" , False,
    [
      "NAME:RLGCCPPInfo",
      "PackageType:=" , "wirebond dieup",
      "LengthUnits:=" , "mm",
      [
        "NAME:Pins"
      ]
    ]
  ], "Example", 100000000)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportConvergence

For a given variation, exports convergence data (max mag delta S, E, freq) to *.conv file.

UI Access	**Results** > **Solution Data**. Select **Convergence** tab and click **Export**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Setup*></td><td>String</td><td>Setup name.</td></tr><tr><td><*VariationKeys*></td><td>String</td><td>The variation. Pass empty string for the current nominal variation.</td></tr><tr><td><*FilePath*></td><td>String</td><td>Full path to desired *.conv file location.</td></tr><tr><td><*ProbType*></td><td>String</td><td>*Optional.* For Q3D Extractor designs. Valid options are "CG", "DC RL", and "AC RL".</td></tr></table>
Return Value	None.

Python Syntax	ExportConvergence(<*Setup*>, <*VariationKeys*>, <*FilePath*> <ProbType>)
Python Example	`oDesign.ExportConvergence('Setup1', 'x_size = 2mm', 'C:/Project12_Q3DDesign1.conv', 'CG')`

ExportDataset

Exports a dataset to a named file. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Export**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*datasetFileFullPath*></td><td>String</td><td>The full path to the file.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportDataset (<*datasetFileFullPath*>)
Python Example	`oProject.ExportDataset('e:/tmp/dsdata.txt')` `oDesign.ExportDataset('e:/tmp/dsdata.txt')`

ExportForHSpice

Exports matrix solution data to a file in a format suitable for HSpice analysis. Available only for Driven Terminal solution types with ports. Output in an appropriate format will be generated for each of the non-empty file names provided.

UI Access	N/A

Parameters

Name	Type	Description
<*DesignVariation*>	String	Desgin variation key.
<*SoluSelections*>	Array	Array of selected solutions.
<*SpiceType*>	Integer	1 - HSpice.
<*BandWidth*>	Integer	0 - Low (narrow) band width.
<*FWSFile*>	String	Full wave spice file name.
<*LumpedElementFile*>	String	Lumped element file name.
<*PoleZeroSpiceFile*>	String	Pole zero spice file name.
<*PoleZeroMatlabFile*>	String	Pole zero Matlab file name.
<*PartialFractionFile*>	String	Partial fraction file name.
<*FittingError*>	Double	The accuracy to use in fitting the pole zero model.
<*MinimumOrder*>	Integer	Minimum number of poles in rational function expansion.
<*MaximumOrder*>	Integer	Maximum number of poles in rational function expansion.
<*UseCommonGround*>	Integer	Optional. 1 to use common ground, 0 otherwise.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*EnforcePassivity*>	Integer	Optional. 1 to enforce passivity, 0 otherwise.
Return Value	None.		

Python Syntax	ExportForHSpice(<*DesignVariation*>, <*SoluSelections*>, <*SpiceType*>, <*BandWidth*>, <*FWSFile*>, <*LumpedElementFile*>, <*PoleZeroSpiceFile*>, <*PoleZeroMatlabFile*>, <*PartialFractionFile*>, <*FittingError*>, <*MinimumOrder*>, <*MaximumOrder*>, [Optional <*UseCommonGround*>], [Optional <*EnforcePassivity*>])

Python Example	`oModule.ExportForHSpice("width='2in'",`
	`["Setup1:Sweep1"], 1, 0,`
	`"c:\mydir\Sweep1.fws", "", "", "", "",`
	`.005, 20, 200)`

ExportForHSpice (Layout Editor)

Use: Export matrix solution data to an HSpice subcircuit.

Command: None

Syntax: ExportForHSpice <SolutionVariationKey> <SolnSelectionArray> <Spice Type> <Bandwidth> <FullWaveSpiceFilename> <LumpedElementFilenme> <Unused> <Unused> <PartialFractionFilename> <Fitting Error> <Unusedint> <MaxOrder> <UseCommonGround> <DoPassivityCheck>

Return Value: None

Parameters: `<SolutionVariationKey>`

Type: string

<SolnSelectionArray>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: array

<Spice Type: 1=HSPICE>

Type: int

<Bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: int

<FullWaveSpiceFilename>

Type: string

<LumpedElementFilenme>

Type: string

<Unused>

Type: string

<Unused>

Type: string

<PartialFractionFilename>

Type: string

<Fitting Error>

Type: double

<Unusedint>

Type: int

<MaxOrder>

Type: int

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<UseCommonGround [Default: FALSE]>

Type: boolean

<DoPassivityCheck [Default: FALSE]>

Type: boolean

ExportForSpice (Layout Editor)

Use: Export matrix solution data to a file in the given spice format.

Command: None

Syntax: ExportForSpice <SolutionVariationKey> <SolnSelectionArray> <Spice Type> <Bandwidth> <FullWaveSpiceFilename> <LumpedElementFilenme> <Unused> <Unused> <PartialFractionFilename> <FittingError> <MaxOrder> <UseCommonGround> <DoPassivityCheck>

Return Value: None

Parameters: `<SolutionVariationKey>`

Type: string

<SolnSelectionArray>

Type: array

<Spice Type: 0=PSpice, 2=MaxwellSpice, 3=Spectre>

Type: int

<Bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: int

<FullWaveSpiceFilename>

Type: string

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<LumpedElementFilenme>

Type: string

<Unused>

Type: string

<Unused>

Type: string

<PartialFractionFilename>

Type: string

<FittingError [Default:0.5]>

Type: double

<MaxOrder [Default:200]>

Type: int

<UseCommonGround [Default:FALSE]>

Type: boolean

<DoPassivityCheck [Default:FALSE]>

Type: boolean

ExportMeshStats

Exports mesh statistics to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<*setupName*>	String	Name of the solution setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*variationString*>	String	Name of the variation. An empty string is interpreted as the current nominal variation.
	<*filePath*>	String	Full file path, including extension *.ms.
	<*overwriteIfExists*>	Boolean	(Optional). Defaults to True. If True, script overwrites any existing file of the same name. If False, no overwrite.
Return Value	None.		

Python Syntax	ExportMeshStats(<*setupName*>, <*variationString*>, <*filePath*>, <*overwriteIfExists*>)
Python Example	`oDesign.ExportMeshStats ("Setup1", "radius=3mm", "C:\mydir\meshstats.ms", True)`

ExportNetworkData

Exports matrix solution data to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<*DesignVariationKey*>	String	Design variation key. Pass empty string for the current nominal variation.
	<*SolnSelectionArray*>	Array	Array of selected solutions. `[<SolnSelector>,<SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating sweep.
	<*SolnSelector*>	String	Solution setup name and solution name, separated by a colon.
	<*FileFormat*>	Integer	File format value.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		2 : Tab delimited spreadsheet format (.tab) 3 : Touchstone (.sNp) 4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<OutFile>	String	Full path to the file to write out.
<FreqsArray>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<DoRenorm>	Boolean	For Touchstone format only. Specifies whether to renormalize the data before export.
<RenormImped>	Double	For Touchstone format only. Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<DataType>	Array	Optional. Type: "S", "Y", or "Z". The matrix to export.
<pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<ComplexFormat>	Integer	Optional. Type: "0", "1", or "2" The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.
<Precision>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
<UseExportFreqs>	Boolean	Specifies whether to use export frequencies.
<IncludeGammaComments>	Boolean	Touchstone only. Specifies whether to include Gamma and Imped-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ance comments.
	<*SupportNonStdExport*>	Boolean	Specifies whether to support non-standard Touchstone extensions for mixed reference impedance.
Return Value	None.		

Python Syntax	ExportNetworkData(<*DesignVariationKey*>, <*SolnSelectionArray*>, <*SolnSelector*>, <*FileFormat*>, <*OutFile*>, <*FreqsArray*>, <*DoRenorm*>, <*RenormImped*>, <*DataType*>, <*pass*>, <*ComplexFormat*>, <*Precision*>, <*UseExportFreqs*>, <*IncludeGammaComments*>, <*SupportNonStdExport*>)
Python Example	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

ExportNetworkData

Exports matrix solution data to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<*DesignVariationKey*>	String	Design variation key. Pass empty string for the current nominal variation.
	<*SolnSelectionArray*>	Array	Array of selected solutions. `[<SolnSelector>,<SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		sweep.
<SolnSelector>	String	Solution setup name and solution name, separated by a colon.
<FileFormat>	Integer	File format value. 2 : Tab delimited spreadsheet format (.tab) 3 : Touchstone (.sNp) 4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<OutFile>	String	Full path to the file to write out.
<FreqsArray>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<DoRenorm>	Boolean	For Touchstone format only. Specifies whether to renormalize the data before export.
<RenormImped>	Double	For Touchstone format only. Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<DataType>	Array	Optional. Type: "S", "Y", or "Z". The matrix to export.
<pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<ComplexFormat>	Integer	Optional. Type: "0", "1", or "2" The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*Precision*>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
	<*UseExportFreqs*>	Boolean	Specifies whether to use export frequencies.
	<*IncludeGammaComments*>	Boolean	Touchstone only. Specifies whether to include Gamma and Impedance comments.
	<*SupportNonStdExport*>	Boolean	Specifies whether to support non-standard Touchstone extensions for mixed reference impedance.
Return Value	None.		

Python Syntax	ExportNetworkData(<*DesignVariationKey*>, <*SolnSelectionArray*>, <*SolnSelector*>, <*FileFormat*>, <*OutFile*>, <*FreqsArray*>, <*DoRenorm*>, <*RenormImped*>, <*DataType*>, <*pass*>, <*ComplexFormat*>, <*Precision*>, <*UseExportFreqs*>, <*IncludeGammaComments*>, <*SupportNonStdExport*>)
Python Example	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

ExportNMFData

Exports s-parameters in neutral file format (*.nmf).

UI Access	**Results** > **Solution Data**. On the **Matrix** tab, click **Export** tab. Then click **Export** > **S-parameter**. Select Neutral Model format and export.		
Parameters	Name	Type	Description
	<*sourceName*>	Array	String names of solutions. Use more than one array entry for a combined Interpolating sweep.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<outputFileName>	String	Full path of output file, including *.nmf extension.
	<reduceMat>	String	Either "Original" or name of reduced matrix.
	<refImp>	Double	Reference impedence.
	<freqs>	Array	Either "All" or array of doubles representing frequencies.
	<variationKey>	String	Design variation key. Pass empty string for the current nominal variation.
	<displayFormat>	String	"MagPhase", "RealImag", "DbPhase".
	<pass>	Integer	Optional. The pass to export. This is ignored if the *sourceName* is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
	<ComplexFormat>	Integer	Optional. The format to use for the exported data. Options are 0 = Magnitude/Phase 1= Real/Imaginary. 2= db/Phase.
	<Precision>	Integer	Optional. Number of digits precision. If not specified, the default is 6.
Return Value	None.		

Python Syntax	ExportNMFData (<*sourceName*>, <*outputFileName*>,<*reduceMat*>, <*refImp*>, <*freqs*>,<*variationKey*>, <*displayFormat*>, <*pass*>, <*ComplexFormat*>, <*Precision*>)
Python Example	`oDesign.ExportNMFData("Setup2 : LastAdaptive", "C:/outputs/data.nmf, "Original", 50, [10000000, 20000000, 30000000, 40000000, 50000000, 60000000, 70000000, 80000000, 90000000, 100000000], "", "MagPhase", 0, 1, 11)`

ExportProfile

Exports a solution profile to file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SetupName*></td><td>String</td><td>Text of the design's notes.</td></tr><tr><td><*VariationString*></td><td>String</td><td>Variation name. Pass empty string for the current nominal variation.</td></tr><tr><td><*filePath*></td><td>String</td><td>Full file path, including extension *.prof.</td></tr><tr><td><*probType*></td><td>String</td><td>*Optional.* For Q3D Extractor designs. Valid options are "CG", "DC RL", and "AC RL".</td></tr></table>
Return Value	None.

Python Syntax	ExportProfile(<*SetupName*>, <*VariationString*>, <*filePath*>, <*probType*>)
Python Example	`oDesign.ExportProfile('Setup1', 'radius=3mm', 'C:\Files\Profile.prof', 'CG')`

ExportReport

Exports a report to a data file.

> **Note:**
>
> The ExportReport script command has been replaced by the script command ExportToFile. ExportReport remains in order to retain backward compatibility for existing scripts, but it is strongly recommended that you now use ExportToFile.

UI Access	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report to be exported</td></tr><tr><td><*FileName*></td><td>String</td><td>Full path of the exported report file name (with extension)</td></tr></table>
Return Value	None

Python Syntax	ExportReport(<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ExportReport(` `"Plot1", "c:\report1.dat")`

ExportRLGCData

Exports matrix solution data to a file.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignVariationKey*></td><td>String</td><td>Design variation key. Pass empty string for the current nominal variation.</td></tr><tr><td><*SolnSelectionArray*></td><td>Array</td><td>Array of selected solutions. `[<SolnSelector>, <SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating sweep.</td></tr><tr><td><*SolnSelector*></td><td>String</td><td>Solution setup name and solution name, separated by a colon.</td></tr><tr><td><*FileFormat*></td><td>Integer</td><td>File format value. 2 : Tab delimited spreadsheet format (.tab) 3 : Touchstone (.sNp)</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<OutFile>	String	Full path to the file to write out.
<FreqsArray>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<DoRenorm>	Boolean	Specifies whether to renormalize the data before export.
<RenormImped>	Double	Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<DataType>	String	Optional. Type: "S", "Y", or "Z". The matrix to export.
<pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<ComplexFormat>	Integer	Optional. Type: "0", "1", or "2" The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.
<Precision>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
<UseExportFreqs>	Boolean	Specifies whether to use export frequencies.
<IncludeGammaComments>	Boolean	Specifies whether to include Gamma and Impedance comments.
<SupportNonStdExport>	Boolean	Specifies whether to support non-standard Touchstone extensions for mixed reference impedance.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportRLGCData(<*DesignVariationKey*>, <*SolnSelectionArray*>, <*SolnSelector*>, <*FileFormat*>, <*OutFile*>, <*FreqsArray*>, <*DoRenorm*>, <*RenormImped*>, [Optional <*DataType*>], [Optional <*pass*>], [Optional <*ComplexFormat*>], [Optional <*Precision*>], [Optional <*UseExportFreqs*>], [Optional <*IncludeGammaComments*>], [Optional <*SupportNonStdExport*>])
Python Example	`oModule.ExportRLGCData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

Generate Mesh

If any mesh operations have been defined but not yet performed in the current variation for the specified solution setups, they will be applied to the current mesh. If necessary, an initial mesh will be computed first. No further analysis will be performed.

UI Access	[**HFSS 3D Layout**] > **Analysis Setup** > **Generate Mesh**.
Parameters	Name | Type | Description <*SimulationNames*> | Array | List of simulation names.
Return Value	Integer: • **0** – Success • **Else** – Failure

Python Syntax	GenerateMesh(<*SimulationNames*>)
Python Example	`oDesign.GenerateMesh(['Setup1','Setup2','Setup3'])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChildNames [Design]

Returns the names of the design's child objects.

UI Access	N/A
Parameters	Name | Type | Description *<type>* | String | Optional. Valid options are "Module", "Editor", "Variable", and "Design Data". Default returns Module names.
Return Value	Names of children for the queried object (i.e., if "Design Data" is entered, **Return Value** will be ['Variables', 'Datasets', 'Design Settings']).

#160;

Python Syntax	GetChildNames (*<type>*)
Python Example	`oDesign.GetChildNames("Design Data")`

GetChildObject [Design]

Returns the design's child objects.

UI Access	N/A
Parameters	Name | Type | Description *<path>* | String | The path may include multiple generations (for example, "designObject/moduleObj/SetupObject"). See: Object Path.
Return Value	A child object if one is found. Otherwise script error.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetChildObject(<*path*>)

Python Example	Optimetrics Example:
	`oDesign = oProject.GetActiveDesign()`
	`oOptimModule = oDesign.GetChildObject("Optimetrics")`
	`oEditor = oDesign.GetChildObject("3D Model")`
	`oBox =  oDesign.GetChildObject("3D Model/Box1")`
	`oRpt = oDesign.GetChildObject("Results/S Parameter Plot 1")`
	`oVariable= oDesign.GetChildObject("Offset")`

GetChildTypes [Design]

Returns the design's child object types.

UI Access	N/A
Parameters	None.
Return Value	String: "Module", "Editor", or "Variable"

Python Syntax	GetChildTypes()
Python Example	`oDesign.GetChildTypes()`

GetDesignID

Returns the unique identification number of the active design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Not Available
Parameters	None
Return Value	String indicating the unique identification number of the active design.

Python Syntax	GetDesignID()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetDesignID()`

GetDesignName

Returns the name of the active design.

UI Access	N/A
Parameters	None.
Return Value	String indicating the name of the active design.

Python Syntax	GetDesignName()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetDesignName()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDesignType

Returns the design type of the active design.

UI Access	N/A
Parameters	None.
Return Value	String indicating the design type of the active design ("Circuit Design", "Circuit Netlist","EMIT", "HFSS 3D Layout Design", "HFSS", "HFSS-IE", "Icepak", "Maxwell 2D", "Maxwell 3D", "Q2D Extractor", "Q3D Extractor", "RMxprt", or "Twin Builder").

Python Syntax	GetDesignType()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetDesignType()`

GetDesignValidationInfo

Returns the design's validation information.

UI Access	N/A
Parameters	Name | Type | Description <*validationResultFilePath*> | String | Full path for validation file, including *.xml extension.
Return Value	Boolean: • **0** – Failure (validation not done) • **1** – Success (results written to specified location; see file format below)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDesignValidationInfo(<*validationResultFilePath*>)
Python Example	`oDesign.GetDesignValidationInfo('E:/MyProjects/Validation.xml')`

GetEditor

Use: Returns an interface to the editor requested IF the PropServer behind the PropHost is contained within that type of editor.

Command: None

Syntax: GetEditor(<editorName>)

Return Value: String

Python Syntax	GetEditor(<editorName>)
Python Example	`oEditor = GetEditor("SchematicEditor")`

GetEditor (Layout Editor)

Use: Get the named editor without activating it.

Command: None

Syntax: GetEditor <editorName object_variable>

Return Value: String

Parameters: <editorName>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

<object_variable>

Type: string

GetGeometryIdsForNetLayerCombinations

Returns ID numbers of all faces and edges of the active design related to the current target combination of nets and layers.

> **Note:** Intended to support CreateFieldPlot.

UI Access	N/A
Parameters	Name / Type / Description *<netName>* / String / Net's name. *<layerName>* / String / Layer's name. *<setupName>* / String / Name of the setup.
Return Value	Array of strings indicating face and edge ID numbers and net/layer combination they are assigned to.

Python Syntax	GetGeometryIdsForNetLayerCombinations ()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetGeometryIdsForAllNetLayerCombinations ("<no-net>","Trace","HFSS Setup : Last Adaptive")`
Example with Variables	`oProject = oDesktop.SetActiveProject("Spiral_Inductor_Microstrip")` `oDesign = oProject.SetActiveDesign("Spiral")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`res = oDesign.GetGeometryIdsForNetLayerCombination("<no-net>","Trace","HFSS Setup : Last Adaptive")` `res = ['Surface', 'FacesList', '21', '22']`

GetGeometryIdsForAllNetLayerCombinations

Returns ID numbers of all faces and edges of the active design for all possible combinations of nets and layers the user can choose.

> **Note:** Intended to support CreateFieldPlot.

UI Access	N/A
Parameters	Name | Type | Description *<setupName>* | String | Name of the setup.
Return Value	Array of strings indicating face and edge ID numbers.

Python Syntax	GetGeometryIdsForAllNetLayerCombinations ()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetGeometryIdsForAllNetLayerCombinations("HFSS Setup : Last Adaptive")`
Example with Variables	`oProject = oDesktop.SetActiveProject("Spiral_Inductor_Microstrip")` `oDesign = oProject.SetActiveDesign("Spiral")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`res = oDesign.GetGeometryIdsForAllNetLayerCombinations("HFSS Setup : Last Adaptive")`
	
	`res = ['PlotGeomInfo for <no-net>/<no-layer> (net/layer combination):', 'Surface', 'FacesList', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '43', '44', '45', '46', '47', '48', '49', '50', '51', '52', '53', '54', '55', '56', '57', 'PlotGeomInfo for <no-net>/AirbridgeMetal (net/layer combination):', 'Surface', 'FacesList', '14', 'PlotGeomInfo for <no-net>/BottomGround (net/layer combination):', 'Surface', 'FacesList', '29', 'PlotGeomInfo for <no-net>/Trace (net/layer combination):', 'Surface', 'FacesList', '21', '22']`

GetManagedFilesPath

Get the path to the project's results folder.

UI Access	N/A
Parameters	None.
Return Value	String containing path where project results are located.

Python Syntax	oDesign.GetManagedFilesPath()
Python Example	`oDesign.GetManagedFilesPath()`

GetModule

Returns the IDispatch for the specified module.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*modulename*></td><td>String</td><td>One of the following: - Excitations Commands – "Excitations" or "Boundaries" - Field Overlays Module – "FieldsReporter" - Optimetrics Module – "Optimetrics" - Output Variable Module – "OutputVariable" - Radiation Setup Manager Module – "RadiationSetupMgr" - Reporter Module – "ReportSetup" - Simulation Setup Commands – "SolveSetups" - User Defined Documents Module – "UserDefinedDocuments" - User Defined Solutions Module – "UserDefinedSolutionModule" </td></tr></table>
Return Value	Module IDispatch

Python Syntax	GetModule (<*modulename*>)
Python Example	`oModule = oDesign.GetModule("SimSetup")`

GetModule (Layout Editor)

Returns the IDispatch for the specified module.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*modulename*></td><td>String</td><td>One of the following: - Excitations Commands – "Excitations" or "Boundaries" - Field Overlays Module – "FieldsReporter" - Optimetrics Module – "Optimetrics" - Output Variable Module – "OutputVariable" - Radiation Setup Manager Module – "RadiationSetupMgr" - Reporter Module – "ReportSetup" - Simulation Setup Commands – "SolveSetups" - User Defined Documents Module – "UserDefinedDocuments" - User Defined Solutions Module – "UserDefinedSolutionModule" </td></tr></table>
Return Value	Module IDispatch

Python Syntax	GetModule (<*modulename*>)
Python Example	`oModule = oDesign.GetModule("Cavities")`

GetModule (RadiationSetupMgr)

Returns the IDispatch for the specified module.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*modulename*></td><td>String</td><td>One of the following:</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Excitations Commands – "Excitations" or "Boundaries" • Field Overlays Module – "FieldsReporter" • Optimetrics Module – "Optimetrics" • Output Variable Module – "OutputVariable" • Radiation Setup Manager Module – "RadiationSetupMgr" • Reporter Module – "ReportSetup" • Simulation Setup Commands – "SolveSetups" • User Defined Documents Module – "UserDefinedDocuments" • User Defined Solutions Module – "UserDefinedSolutionModule"
Return Value	Module IDispatch		

Python Syntax	GetModule (<*modulename*>)
Python Example	`oModule = oDesign.GetModule("SimSetup")`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetName (Layout Editor)

Get the name of the active design.

Command: None

Syntax: GetName <namestring_variable>

Return Value: String

Parameters: `<namestring_variable>`

Type: string

GetNoteText

Returns the text of the note attached to a design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	String: text of the design note.

Python Syntax	GetNoteText()
Python Example	`oDesign.GetNoteText()`

GetObjPath [Design]

Obtains the path to the design.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the design.

Python Syntax	GetObjPath()
Python Example	`oDesign.GetObjPath()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetOutputVariableValue

Returns the double value of an output variable. Only expressions that return a double value are supported. The expression is evaluated only for a single point.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OutputVarName></td><td>String</td><td>Name of the output variable.</td></tr>
<tr><td><IntrinsicVariation></td><td>String</td><td>A set of intrinsic variable value pairs to use when evaluating the output expression.
HFSS example:
<code>"Freq='20GHz' Theta='20deg' Phi='30deg'"</code></td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution as listed in the output variable UI. For example, <code>"Setup1 : Last Adaptive"</code>.</td></tr>
<tr><td><ReportType></td><td>String</td><td>The name of the report type as seen in the output variable UI.
Possible values in Layout/Schematic Editor are:
"Standard"
"Load Pull"
"Constellation"
"Data Table"
"Eye Diagram"
"Statistical"</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated. Can be empty.
<code>["Context:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Double value of the output variable.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetOutputVariableValue(<*OutputVarName*>, <*IntrinsicVariation*>, <*SolutionName*>, <*ReportTypeName*>, <*ContextArray*>)
Python Example	`Val = oDesign.GetOutputVariableValue("test","Freq = '20Ghz' Theta='20deg' Phi='30deg'", "TR", "Standard", [])`

GetOutputVariables

Returns the list of output variables.

UI Access	N/A
Parameters	None.
Return Value	Array containing all output variables.

Python Syntax	GetOutputVariables()
Python Example	`oDesign.GetOutputVariables()`

GetProject

Returns to the active design's project or, if a project name is specified in the parentheses, changes focus to the specified project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ProjectName*></td><td>String</td><td>Name of an open project. Can be empty.</td></tr></table>
Return Value	None.

Python Syntax	GetProject(<*ProjectName*>)
Python Example	`oDesign.GetProject('Project2')`

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

GetProperties

Gets a list of all the properties belonging to a specific <PropServer> and <PropTab>. This can be executed by the oProject, oDesign, or oEditor variables.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PropTab*></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<*PropServer*>	String	An object identifier, generally returned from another script method, such as CompInst@R;2;3
Return Value	Array of strings containing the names of the appropriate properties.		

Python Syntax	GetProperties(<*PropTab*>, <*PropServer*>)
Python Example	`oEditor.GetProperties('PassedParameterTab', 'k')`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Tip: Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PropTab></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:
PassedParameterTab ("Parameter Values")
DefinitionParameterTab (Parameter Defaults")
LocalVariableTab ("Variables" or "Local Variables")
ProjectVariableTab ("Project variables")
ConstantsTab ("Constants")
BaseElementTab ("Symbol" or "Footprint")
ComponentTab ("General")
Component("Component")
CustomTab ("Intrinsic Variables")
Quantities ("Quantities")
Signals ("Signals")</td></tr>
<tr><td><PropServer></td><td>String</td><td>An object identifier, generally returned from another script method, such as <code>Comp:Inst@R;2;3</code></td></tr>
<tr><td><PropName></td><td>String</td><td>Name of the property.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>String value of the property.</td></tr>
</table>

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)
Python Example	`selectionArray = oEditor.GetSelections()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	for k in selectionArray: val = oEditor.GetPropertyValue("PassedParameterTab", k, "R") ...

GetPropNames [Design]

Returns array containing names of property. For designs, always returns an empty array because the design has no property.

UI Access	N/A
Parameters	Name: *<includeReadOnly>* Type: Boolean Description: (Optional). • True - include read only property. • False - do not include read only property.
Return Value	Empty array.

Python Syntax	GetPropNames(<*includeReadOnly*>)
Python Example	`oDesign.GetPropNames()`

GetPropValue [Design]

Returns the property value for the active design object, or specified property values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*propPath*></td><td>String</td><td>A child object's property path. See: Object Property Script Function Summary.</td></tr></table>
Return Value	The property value (integer, string, or array of strings) of the specified child object.

Python Syntax	GetPropValue(<*propPath*>)

Python Example	`oDesign.GetPropValue('offset/SIValue')`
	`oDesign.GetPropValue('Results/S Parameter Plot 1/Display Type')`
	`oDesign.GetPropValue('Results/S Parameter Plot 1/Display Type/Choices')`

GetVariables

Returns a list of all defined variables. To get a list of project variables, execute this command using `oProject`. To get a list of local variables, use `oDesign`.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the variables.

Python Syntax	GetVariables ()
Python Example	`oProject.GetVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oDesign.GetVariables()`

GetVariableValue

Gets the value of a single specified variable. To get the value of project variables, execute this command using `oProject`. To get the value of local variables, use `oDesign`.

UI Access	N/A
Parameters	Name | Type | Description <*VarName*> | String | Name of the variable to access.
Return Value	String represents the value of the variable.

Python Syntax	GetVariableValue(<*VarName*>)
Python Example	`oProject.GetVariableValue("var_name")`

ImportDataset

Imports a dataset from a named file. This can be executed by the oProject, or oDesign variables. The name of the dataset is file-name+index number (e.g., dsdata1) unless the filename ends with a trailing number. When there is a trailing number at the end, we will remove the number and use first unused index. Alternatively, the name of the dataset can be explicitly defined by providing a string as an optional second argument.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Project** > **Datasets** > **Import**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*datasetFileFullPath*></td><td>String</td><td>The full path to the file containing the dataset values. *.tab files recommended (see note below).</td></tr><tr><td><*optionalDatasetName*></td><td>String</td><td>*Optional*. User-defined dataset name.</td></tr></table>
Return Value	None.

Python Syntax	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)

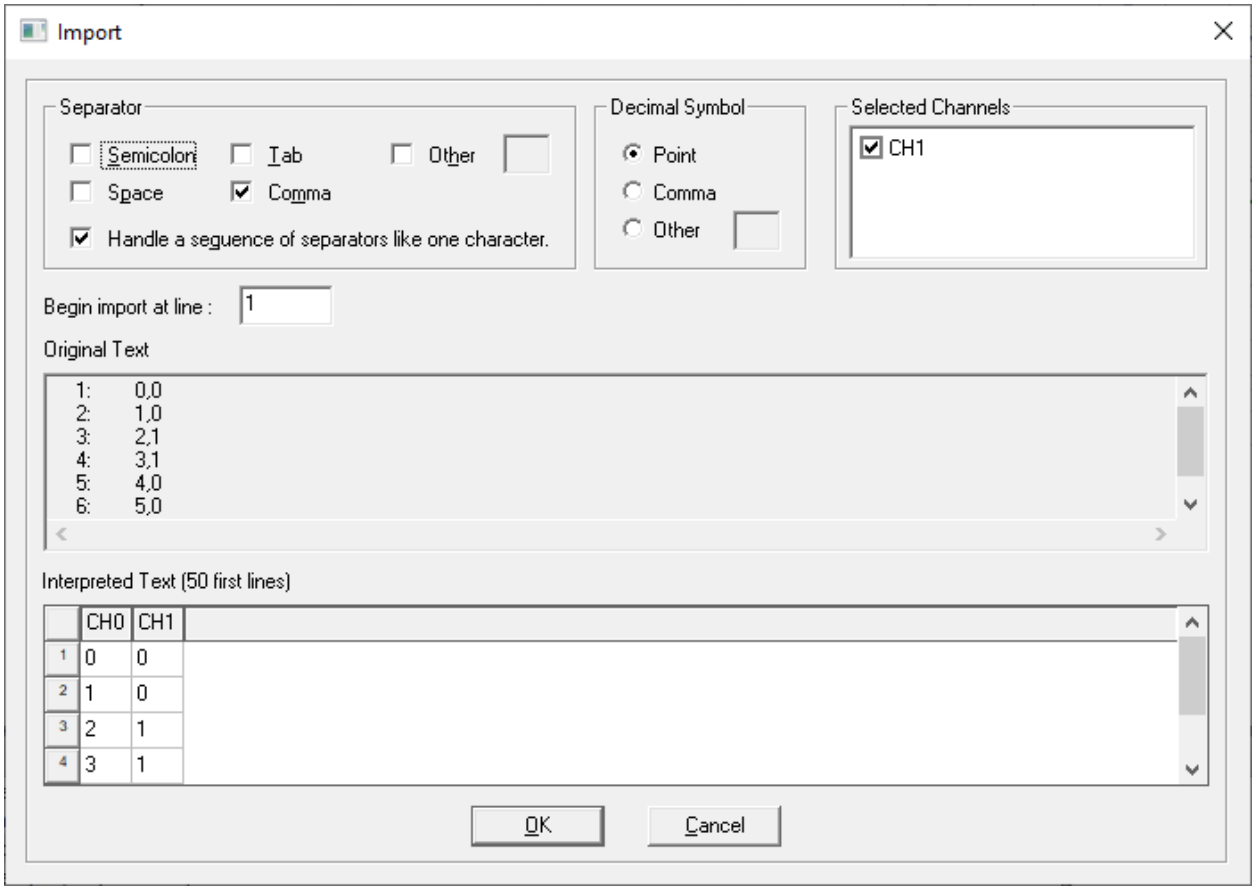
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

Note About File Types:

Tab-delimited or space-delimited files with the extension *.tab are the recommended file type. When using ImportDataset at the Design level, *.tab is the only file type supported.

At the Project level, other file types are supported (for example, *.csv). However, after calling the command, you must configure the file import format manually through the Electronics Desktop GUI by selecting **Project** > **Datasets** and clicking **Import**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



InsertDesign

Inserts a new design in the project. For Circuit and HFSS 3D Layout scripts, the last argument will always be empty.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Project** > **[Insert Circuit Design / Insert Circuit Netlist / Insert HFSS 3D Layout Design]**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignType*></td><td>String</td><td>Design type. Valid types: "Circuit", "Nexxim Circuit", "System" , "Nexxim Net-list", "HFSS3D".</td></tr><tr><td><*DesignName*></td><td>String</td><td>Design name.</td></tr><tr><td><*TechnologyFile*></td><td>String</td><td>The path to the Circuit technology file to be used in the design. Use a pair of empty double quotes ("") for none.</td></tr><tr><td><*SubcircuitID*></td><td>String</td><td>(Optional) Must be preceded by a colon if included along with the Technology File name. No colon is necessary when the subcircuit ID is omitted.</td></tr><tr><td>""</td><td>None</td><td>Empty argument.</td></tr></table>
Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*TechnologyFile*>:<*SubCircuitID*>,")
Python Example	`oProject.InsertDesign("Nexxim Circuit","MyDesign", "C:\\Program Files\\ANSYS Inc\\v261\\AnsysEM", "")`

OverlayCurrents (Layout Editor)

Creates a 3D view with a currents overlay

Command: None

Syntax: OverlayCurrents <solution name>

Return Value: <success>

Type: boolean

Parameters: <solution name>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

OverlayFarField (Layout Editor)

Creates a 3D view with a far field overlay

Command: None

Syntax: OverlayFarField <solution name>

Return Value: <success>

Type: boolean

Parameters: `<solution name>`

Type: string

OverlayMesh (Layout Editor)

Creates a 3D view with a mesh overlay

Command: None

Syntax: OverlayMesh <solution name>

Return Value: <success>

Type: boolean

Parameters: `<solution name>`

Type: string

OverlayNearField (Layout Editor)

Creates a 3D view with a near field overlay

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: None

Syntax: OverlayNearField <solution name>

Return Value: <success>

Type: boolean

Parameters: <solution name>

Type: string

PasteDesign (Design Object)

Pastes a design that has already been copied to the clipboard into another design.

UI Access	**Edit** > **Paste**.
Parameters	Name: <*pasteOption*> — Type: Integer — Description: One of the following: • **0** – Link to the existing design that was copied • **1** – Create a new copy of the design and keep the original layers of the design being copied • **2** – Create a new copy of the design and merge the layers of the design being copied.
Return Value	None.

Python Syntax	PasteDesign(<*pasteOption*>)
Python Example	`oProject.CopyDesign('DesignB')` `oDesign = oProject.SetActiveDesign('DesignA')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oDesign.PasteDesign(1)`

PasteItemCommand (Layout Editor)

Paste tree items, such as Altrasim Solution and Solve Setups pasted to the Analysis Tree, or Frequency Sweeps pasted to the Solve Setup Tree.

Command: Right-click on the Analysis item in the Project tree and select Paste.

Syntax: PasteItemCommand <ItemPath>

Return Value: None

Parameters: `<ItemPath>`

Type: string

PushExcitations

Allows access to computed excitations for transient and linear frequency solutions. The script command can be accessed from three locations.

<table>
<tr><td>UI Access</td><td>

Layout Editor
Schematic Editor
Select a Nexxim solution in a 3D Layout design, right click, and choose Push Excitations.

</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Reference Designation></td><td>String</td><td>"<refdes>"

This argument is empty when excitations are pushed from a Nexxim solution.</td></tr>
<tr><td><PushExcitations Parameters></td><td>Array</td><td>This parameter changes depending on whether the excitations comes from a transient or linear frequency solution. The keyword</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"transient:=" indicates to AEDT which solution generated the excitations. If "transient:=" is present, "CalcThevenin =" and its value are ignored. For a linear frequency solution, use this array: ["NAME:options", "CalcThevenin =" , <true|false>, "Sol:=", "<solution name>"] If CalcThevenin is true, Thevenin's equivalent is calculated. Parameters for the linear frequency solution do not include a Freq argument, so all frequencies from the solution are used. For a transient solution, use: ["NAME:options", "transient:=", ["start:=", <start time>, "stop:=", <stop time>, "maxHarmonics:=", <max harmonics>, "winType:=", <window>, ["widthPct:=", <width percentage>,] ["kaiser:=", <Kaiser value>,] ["correctCoherentGain:=", true]], "Sol:=", "<solution name>"] winType can have the following values: • Rectangular

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Bartlett • Blackman • Hamming • Hanning • Kaiser • Welch • Weber • Lanzcos
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Redo [Design]

Reapplies the last design-level command.

UI Access	**Edit** > **Redo**.
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Redo()
Python Example	`oDesign.Redo()`

Redo (Layout Editor)

Redo the last operation

Command: Edit>Redo

Syntax: Redo

Return Value: None

Parameters: None

RemoveImportData (Layout Editor)

Remove an imported solution

Command: None

Syntax: RemoveImportData <solution_name>

Return Value: None

Parameters: <solution_name>

Type: string

RemoveModelingProperties (Layout Editor)

Remove a modeling property from a design

Command: None

Syntax: RemoveModelingProperties <design>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None

Parameters: <design>

Type: string

RenameDesignInstance

Renames a design instance.

UI Access	Right-click a design instance in the project tree, and then click **Rename** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OldName*></td><td>String</td><td>The current name of the design, which must be the design on which this command is invoked.</td></tr><tr><td><*NewName*></td><td>String</td><td>The new name for the design.</td></tr></table>
Return Value	None.

Python Syntax	RenameDesignInstance (<*OldName*>, <*NewName*>)
Python Example	`oDesign.RenameDesignInstance("Design1", "Design2")`

RenameDesignInstance (Layout Editor)

Rename the design instance

Command: None

Syntax: RenameDesignInstance <oldname> <newname>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Example: oDesign.RenameDesignInstance <oldname> <newname>

RenameImportData (Layout Editor)

Rename an imported solution

Command: None

Syntax: RenameImportData <oldname> <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Example: oDesign.RenameImportData <oldname> <newname>

RenameSource [Interface Source]

Use: Renames an interface source

Syntax: RenameSource (STRING: Interface Source name)

Example: oModule.RenameSource "OldName" "NewName"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ReportTemplates (Layout Editor)

Use: Creates a report using one of the HFSS 3D Layout report templates

Command: None

Syntax: ReportTemplates <template>

Return Value: None

Parameters: `<template>`

Type: string

Example: `oDesign.ReportTemplates <template>`

RunToolkit

Runs a Python toolkit script, applying it to the active design. The toolkit script itself may have prerequisites, such as sweeps defined, or specific model characteristics, such as port definitions.

> **Important:**
>
> Full scripting for cable modeling is not supported and no arguments are allowed in the script. The **RunToolkit** command will not automatically create a cable bundle, but will simply open the **Cable Modeling** dialog box. You will have to manually input your parameters.

UI Access	[**HFSS 3D Layout**] > **Toolkit** > [**Script Name**].		
Parameters	Name	Type	Description
	<LibraryType>	String	Name of the library in which the script is located.
	<ToolkitName>	String	Name of the Python script.
	<ToolkitArg>	Array	Structured array containing arguments required by the toolkit script.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	User-defined solution(s) and/or output(s).

Python Syntax	RunToolkit(<*LibraryType*>, <*ToolkitName*>, <*ToolkitArg*>)
Python Example	`oDesign.RunToolkit("SysLib", "Hearing Aid Compliance", [])`

SetActiveEditor

Sets the active editor.

UI Access	N/A.
Parameters	Name | Type | Description <*EditorName*> | String | Text of the design's notes.
Return Value	Editor object.

Python Syntax	SetActiveEditor(<*EditorName*>)
Python Example	`oDesign.SetActiveEditor('3D Modeler')`

SetActiveEditor (Layout Editor)

Set the active editor to the named one. (Activates window or opens window if necessary).

Command: None

Syntax: SetActiveEditor <editorName> <object_variable>

Return Value: None

Parameters: `<editorName>`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

<object_variable>

Type: string

Example: `oDesign.SetActiveEditor <editorName> <object_variable>`

SetBendGraphicsMode

Activates/deactivates a simplified version of the model rendering for designs with rigid-flex bends.

UI Access	From the **Layout** tab, select **Rigid-Flex** > **Simplified Bend Graphics**.
Parameters	Name: <*SetBendGraphics*> — Type: Integer — Description: **0** – Deactivates **Simplified Bend Graphics**; **1** – Activates **Simplified Bend Graphics**
Return Value	None.

Python Syntax	SetBendGraphics (<*SetBendGraphics*>)
Python Example	`oEditor.SetBendGraphicsMode("1")`

SetDesignMode

Switches between HFSS 3D Layout operating modes.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Edit Layout...**		
Parameters	Name	Type	Description
	<modeName>	String	Enter "**IC**" or "**General**" to change operating modes.
Return Value	None.		

Python Syntax	SetDesignMode(<*modeName*>)
Python Example	`oDesign.SetDesignMode("IC")`

SetDoMeshAssembly

Set the active Mesh fusion properties for each object.

Command: Set Components for Mesh Fusion

Syntax: SetDoMeshAssembly <*MeshAssemblyParameters*>

Return Value: None

Parameters: `<MeshAssemblyParameters>`

Available script command choices for LayoutCompMeshEnvelopeType are:

- “Bounding Box”
- “Conformal”
- “Convex Hull”
- “Zone”

For MeshAssemblyBoundingVolumePadding, users are expected to specify all padding values in sequence.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

1. For Bounding Box type, padding values are expected in the following format:

   [+X, -X, +Y, -Y, +Z, -Z]

2. For Conformal and Convex Hull type, padding values are expected in the following format:

   [+Horizontal, +Vertical, -Vertical]

3. For Zone envelope type, padding values are expected in the following format:

   [+Vertical, - Vertical]

   For all cases, additional values after those expected will be ignored.

Python Examples

Example with Priority Level Enabled, with a section for Component Priority Level is added.

```
oDesign.SetDoMeshAssembly(
  [
    "NAME:AllSettings",
    [
      "NAME:MeshAssembly",
      [
        "NAME:LC1_2",
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ],
      [
        "NAME:LC1_3",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ],
      [
        "NAME:Modal1",
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ]
    ],
    "Component Priority Level:=", ["LC1_1","Modal1","LC1_2","Modal_with_mesh_region1","LC1_
    3","Modal3"]
  ])
```

If a design which had Priority Level set but then disabled, a section of the script records the disabled priority level.

```
oDesign.SetDoMeshAssembly(
  [
    "NAME:AllSettings",
    [
      "NAME:MeshAssembly",
      [
        "NAME:LC1_3",
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ],
      [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:Modal1",
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ]
    ],
    "Priority Components:=" , ["LC1_3","Modal1","Modal_with_mesh_region1"]
  ])
```

If Priority Level is not enabled, the behavior is as before.

```
oDesign.SetDoMeshAssembly(
  [
    "NAME:AllSettings",
    [
      "NAME:MeshAssembly",
      [
        "NAME:Cylinder3",
        [
          "NAME:MeshSetting",
          [
            "NAME:GlobalSurfApproximation",
            "CurvedSurfaceApproxChoice:=", "UseSlider",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "SliderMeshSettings:=" , 7
      ],
      [
        "NAME:GlobalCurvilinear",
        "Apply:=" , False
      ],
      [
        "NAME:GlobalModelRes",
        "UseAutoLength:=" , True
      ],
      "MeshMethod:=" , "AnsoftClassic",
      "UseLegacyFaceterForTauVolumeMesh:=", False,
      "DynamicSurfaceResolution:=", False,
      "UseFlexMeshingForTAUvolumeMesh:=", False,
      "UseAlternativeMeshMethodsAsFallBack:=", True,
      "AllowPhiForLayeredGeometry:=", True
    ],
    "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
  ],
  [
    "NAME:Pentagon_tilted2_1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
      ],
      [
        "NAME:Cone_with_variable1",
        "MeshAssemblyBoundingVolumePadding:=", ["0","0","0","0","0","0"]
    ]
    ],
    [
      "NAME:Priority Components",
      "Cylinder3",
      "Cone_with_variable1",
      "BoxWithMeshRegion1"
    ]
  ])
```

For Layout Component

```
oDesign.SetDoMeshAssembly(
  [
    "NAME:AllSettings",
    [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:MeshAssembly",
      [
        "NAME:LC1_1",
        "LayoutCompMeshEnvelopeType:=", "Zone",
        "MeshAssemblyBoundingVolumePadding:=", ["100","200"]
      ]
    ]
  ])
```

SetFastTransformationForLayoutComponent

Toggles whether layout components in show outlines of objects across layers during view changes. This improves the visualization performance.

UI Access	**Fast Transformation**		
Parameters	Name	Type	Description
	<Layout ComponentName>	String	Name of the Layout Component.
	Boolean	String	True or False
Return Value	None.		

Python Syntax	SetFastTransformationForLayoutComponent ("<LayoutComponentName", <boolean>)
Python Example	`oDesign.SetFastTransformationForLayoutComponent ("LC1_1", False)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetObjectAttributesForLayoutComponent

Sets visualization attributes for layout components in for a named component for layers, nets, and objects.

<table>
<tr><th>UI Access</th><th colspan="3">Object Attributes</th></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Layout ComponentName></td><td>String</td><td>Name of the Layout Component.</td></tr>
<tr><td><ShowDielectric></td><td>Boolean</td><td>True to show dielectrics; false to hide dielectrics</td></tr>
<tr><td><DisplayMode></td><td>Integer</td><td>0 (for layout mode), 1 (for net mode), and 2 (for object mode)</td></tr>
<tr><td><ObjectAttributesinLayerMode></td><td>Array</td><td>List of layers. The three arguments for each layer represent the “Show”, “Wire Frame”, “and “Transparency” options for that layer in the “Object Attributes” dialog box.</td></tr>
<tr><td><ObjectAttributesinNetMode></td><td>Array</td><td>List of nets. The three arguments for each net represent the “Show”, “Wire Frame”, “and “Transparency” options for that net in the “Object Attributes” dialog box.</td></tr>
<tr><td><ObjectAt-tributesinObjectMode></td><td>Array</td><td>List of objects. The three arguments for each object represent the “Show”, “Wire Frame”, “and “Transparency” options for that object in the “Object Attributes” dialog box.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

Python Syntax	SetObjectAttributesForLayoutComponent (<*LayoutComponentName*>, <*ShowDielectric*>, <*DisplayMode*>, <*ObjectAttributesinLayerMode*> <*ObjectAttributesinNetMode*>, <*ObjectAttributesinObjectMode*>)

Python Example	`oDesign.SetObjectAttributesForLayoutComponent (`
	`  [`
	`  "Name:="  , "LC1_1",`
	`  "ShowDielectric:="  , False,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "DisplayMode:=" , 2,
    [
      "NAME:ObjectAttributesInLayerMode",
      "GND_1_L2:="            , [True,True,0],
      "GND_2_L4:="            , [True,True,0],
      "GND_3_L6:="            , [True,True,0],
      "GND_4_L9:="            , [True,True,0],
      "GND_5_L11:="           , [True,True,0],
      "Inner_Layer_3_L10:="   , [True,True,0],
      "Top_L1:="              , [True,True,100],
      "VCC1_L7:="             , [True,True,0],
      "VCC2_L8:="             , [True,True,0]
    ],
    [
      "NAME:ObjectAttributesInNetMode",
      "GND:="         , [True,True,29],
      "VCC:="         , [True,True,0],
      "neg:="                 , [True,True,67],
      "pos:="                 , [True,True,67]
    ],
    [
      "NAME:ObjectAttributesInObjectMode",
      "line_1:="              , [True,True,0],
      "line_2:="              , [True,True,0],
      "line_3:="              , [True,True,100],
      "line_4:="              , [True,True,100],
      "rect_6:="              , [True,True,0],
      "rect_17:="             , [True,True,0],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "rect_18:="             , [True,True,0],
        "rect_19:="             , [True,True,0],
        "rect_20:="             , [True,True,0],
        "rect_21:="             , [True,True,0],
        "rect_22:="             , [True,True,0],
        "line_24:="             , [True,True,100],
        "via_0:="               , [True,True,100],
        "via_1:="               , [True,True,100],
        "via_2:="               , [True,True,100],
        "via_3:="               , [True,True,100]
    ]
])
```

SetPropValue [Design]

Sets the property value for the active property object.

UI Access	Edit properties on Project Tree objects.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*propPath*></td><td>String</td><td>A child object's property path. See: Object Property Script Function Summary.</td></tr><tr><td><*Value*></td><td>String</td><td>New property value.</td></tr></table>
Return Value	Boolean: • **True** - property found and the new value is valid. • **False** - property not found.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SetPropValue(<*propPath*>, <*Value*>)
Python Example	`oDesign.SetPropValue("offset", "12mm")` `oDesign.SetPropValue("Results/S Parameter Plot 1/Display Type", "Rectangular Plot")`

SetPropertyValue

Sets the value of a single property belonging to a specific PropServer and PropTab. This function is available with the Project, Design or Editor objects, including definition editors.This is not supported for properties of the following types: ButtonProp, PointProp, V3DPointProp, and VPointProp. Only the ChangeProperty command can be used to modify these properties.

Use the script recording feature and edit a property, and then view the resulting script entry or use GetPropertyValue for the desired property to see the expected format.

UI Access	N/A

Parameters

Name	Type	Description
<*propTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Signals ("Signals")
	<*propServer*>	String	An object identifier, generally returned from another script method, such as Comp⊦nst@R;2;3
	<*propName*>	String	Name of the property.
	<*propValue*>	String	The value for the property
Return Value	None.		

Python Syntax	SetPropertyValue(<*propTab*>, <*propServer*>, <*propName*>, <*propValue*>)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

SetShowLayoutForLayoutComponent

Layout visualization, rather than bounding box, for a Layout Component in design.

UI Access	**Edit Layout...**		
Parameters	Name	Type	Description
	<*Layout ComponentName*>	String	Name of the Layout Component.
	Boolean	String	True or False
Return Value	None.		

Python Syntax	SetShowLayoutForLayoutComponent ("<LayoutComponentName", <boolean>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesign.SetShowLayoutForLayoutComponent ("LC1_1", True)`

SetVariableValue

Sets the value of a variable. To set the value of a project variable, execute this command using `oProject`. To set the value of a local variable, use `oDesign`.

UI Access	N/A
Parameters	Name | Type | Description <*VarName*> | String | Variable name. <*VarValue*> | Value | New value for the variable.
Return Value	None.

Python Syntax	SetVariableValue (<*VarName*>, <*VarValue*>)
Python Example	`oProject.SetVariableValue('$Var1', '3mm')`

Undo [Design]

Cancels the last design-level command.

UI Access	**Edit** > **Undo**
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Undo()
Python Example	`oDesign.Undo()`

Undo (Layout Editor)

Undo the last operation

Command: Edit>Undo

Syntax: Undo

Return Value: None

Parameters: None

Example: `oDesign.Undo`

ValidateCircuit (Layout Editor)

Use: Validates a HFSS 3D Layout design for consistency

Command: None

Syntax: ValidateCircuit

Return Value: Boolean:

- `1` - design is valid
- `0` - design is not valid

Parameters: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: oDesign.ValidateCircuit

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

8 - 3D Modeler Editor Script Commands

3D Modeler commands should be executed by the "3D Modeler" editor:

```
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.<CommandName>
```

Conventions Used in this Chapter:

<AttributesArray>

<AttributesArray> takes the following structure:

```
["NAME:Attributes",
     "Name:=", <string>,
     "Flags:=", <string>,
     "Color:=", <string>,
     "Transparency:=", <value>,
     "PartCoordinateSystem:=", <string>,
     "UDMId:=", <string>,
     "MaterialValue:=", <string>,
     "SurfaceMaterialValue:=", <string>,
     "Solveinside:=", <boolean>,
     "ShellElement:=", <boolean>,
     "ShellElementThickness:=", <string>,
     "IsMaterialEditable:=", <boolean>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"UseMaterialAppearance:=", <boolean>,
"IsLightweight:=", <boolean>]
```

Where:

- **Flags** – Takes a string containing "NonModel" and/or "Wireframe", separated by the # character. For example, "NonModel#Wireframe".
- **Color** – Takes a string containing an RGB triple, formatted as "(<*RGB*>)". For example, "(255 255 255)".
- **Transparency** – Takes a value between 0 and 1.
- **PartCoordinateSystem** – Orientation of the primitive. The name of one of the defined coordinate systems should be specified.
- **UDMId** – Takes a string containing an ID.
- **MaterialValue** – Takes a string of the material name.
- **SurfaceMaterialValue** – Takes a string of the surface material name.
- **Solveinside** – Takes a boolean value.
- **ShellElement** – Takes a boolean value specifying whether or not a shell element is present.
- **ShellElementThickness** – Takes a string containing the shell element thickness. If element is not present, pass empty string.
- **IsMaterialEditable** – Takes a boolean value.
- **IsLightweight** – Takes a boolean value.

<SelectionsArray>

<SelectionsArray> typically takes the following structure:

```
[NAME:Selections", "Selections:=", <string>]
```

Where:

- **Selections** – Takes a comma-separated list of parts on which to perform the action. For example, "Rect1, Rect2, Rect3".

In some cases, <SelectionsArray> takes additional parameters:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:Selections",
  "AllowRegionDependentPartSelectionForPMLCreation:=", <boolean>,
  "AllowRegionSelectionForPMLCreation:=", <boolean>,
  "Selections:=", <string>,
  "NewPartsModelFlag:=" , <string>,
  "UseCurrentCS:=", <boolean>]
```

Where:

- **AllowRegionDependentPartSelectionForPMLCreation** – Takes a boolean value. See individual script for whether this parameter is required.
- **AllowRegionSelectionForPMLCreation** – Takes a boolean value. See individual script for whether this parameter is required.
- **Selections** – Takes a comma-separated list of parts on which to perform the action. For example, "Rect1, Rect2, Rect3".
- **NewPartsModelFlag** – Takes either string "Model" or string "Nonmodel". See individual script for whether this parameter is required.
- **UseCurrentCS** – Takes a boolean value. See individual script for whether this parameter is required. Use GetActiveCoordinateSystem to determine the current CS.

> **Note:**
>
> *Selections* is the only parameter required in *all* 3D Modeler Editor scripts. See individual scripts for additional required parameters.

Organization

3D Modeler editor scripts are organized into the following categories:

Draw Menu Commands

Edit Menu Commands

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Modeler Menu Commands

Cable Modeling Commands

Other oEditor Commands

Draw Menu Commands

Create3DComponent

CreateBondwire

CreateBox

CreateCircle

CreateCone

CreateCutplane

CreateCylinder

CreateEllipse

CreateEquationCurve

CreateEquationSurface

CreateHelix

CreatePoint

CreatePolyline

CreateRectangle

CreateRegion

CreateRegularPolygon

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateRegularPolyhedron

CreateSphere

CreateSpiral

CreateSubregion

CreateTorus

CreateUserDefinedPart

Edit3DComponent

EditPolyline

Get3DComponentDefinitionNames

Get3DComponentInstanceNames

Get3DComponentMaterialNames

Get3DComponentMaterialProperties

Get3DComponentParameters

Insert3DComponent

InsertPolylineSegment

SweepAlongPath

SweepAlongVector

SweepAroundAxis

SweepFacesAlongNormal

SweepFacesAlongNormalWithAttributes

UpdateComponentDefinition

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Create3DComponent

Creates a 3D component.

<table>
<tr><td>UI Access</td><td>Draw > 3D Component Library > Create 3D Component.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><CreateData></td><td>Array</td><td>Structured array containing geometry data:
<pre>["NAME:CreateData",
  "ComponentName:=", "<string>",
  "Company:=", "<string>",
  "Company URL:=", "<string>",
  "Model Number:=","<string>",
  "Help URL:=" , "<string>",
  "Version:=", "<string>",
  "Notes:=", "<string>",
  "IconType:=","<string>",
  "Owner:=", "<string>",
  "Email:=", "<string>",
  "Date:=", "<string>", #Component creation date
  "HasLabel:=", <boolean>,
  "IsEncrypted:=", <boolean>,
  "AllowEdit:=", <boolean>,</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"SecurityMessage:=" , "<string>", "Password:=" , "<string>", "EditPassword:=" , "<string>", "PasswordType:=" , "<string>", "HideContents:=" , <boolean>, "ReplaceNames:=" , <boolean>, #for objects and materials "ComponentOutline:=" , "<string>",#"None" or "Bounding Box" "IncludedParts:=" , [Array of partnames], "HiddenParts:=" , [Array of hidden parts], "IncludedCS:=" , [Array of included CS], "DefaultHandle:=" , "<reference CS>" "IncludedParameters:=" , [Array of included parameters], "IncludedDependentParameters:=", [Array of dependent parameters], "ParameterDescription:=", <array> "IsLicensed:=" , <boolean> "LicensingDllName:=" , "<string>" "VendorComponentIdentifier:=", "<string>" "PublicKeyFile:=" , "<string>"]
	<*DesignData*>	Array	Structured array containing design data:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:DesignData", "Boundaries:=", <array>, "Excitations:=", <array>, "MeshOperations:=", <array>]
	<*FileName*>	String	Full file path to 3D component.
	<*ImageFile*>	Array	Optional. Structured array containing file path of 3D component image: ["NAME:ImageFile", "ImageFile:=", <string>])
Return Value	None.		

Python Syntax	Create3DComponent(<*CreateData*>, <*DesignData*>, <*FileName*>, <*ImageFile*>)
Python Example	oEditor.Create3DComponent(["NAME:CreateData", "ComponentName:=", "Connector", "Company:=", "", "Company URL:=" , "" "Version:=", "1.0", "Notes:=", "", "Owner:=", "", "IconType:=" , "", "Email:=", "", "Date:=", "11:41:01 AM Aug 28, 2024", "HasLabel:=", false, "IsEncrypted:=" , False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "AllowEdit:=" , False,
    "SecurityMessage:=" , "",
    "Password:=" , "",
    "EditPassword:=" , "",
    "PasswordType:=" , "UnknownPassword",
    "HideContents:=" , True,
    "ReplaceNames:=" , True,
    "ComponentOutline:=" , "None",
    "IncludedParts:=", ["Box1", "Cylinder1", "Cone1"],
    "IncludedCS:=", ["RelativeCS1"],
    "DefaultHandle:=", "Global",
    "IncludedParameters:=", ["htcone", "lr", "htcyl", "zs", "radcyl", "xs",
    "$rp", "$con"],
    "IncludedDependentParameters:=", [],
    "ParameterDescription:=", [],
    "IsLicensed:=" , False,
    "LicensingDllName:=" , "",
    "VendorComponentIdentifier:=", "",
    "PublicKeyFile:=" , ""
  ],
  [
    "NAME:DesignData",
    "Boundaries:=", ["PerfE1", "FiniteCond1"],
    "Excitations:=", ["1"],
    "MeshOperations:=", []
  ],
  "C:/tmp/Connector.a3dcomp",
  [
    "NAME:ImageFile", "ImageFile:=", ""
  ])
```

Python Example

Example with an External Circuit:

```
oDesign = oProject.SetActiveDesign("Maxwell3DDesign1")
oEditor = oDesign.SetActiveEditor("3D Modeler")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oEditor.Create3DComponent(
  [
    "NAME:CreateData",
    "ComponentName:=" , "Maxwell3DDesign1",
    "Company:=" , "",
    "Company URL:=" , "",
    "Model Number:=" , "",
    "Help URL:=" , "",
    "Version:=" , "1.0",
    "Notes:=" , "",
    "IconType:=" , "",
    "Owner:=" , "",
    "Email:=" , "",
    "Date:=" , "3:36:11 PM Feb 14, 2025",
    "HasLabel:=" , False,
    "IsEncrypted:=" , False,
    "AllowEdit:=" , False,
    "SecurityMessage:=" , "",
    "Password:=" , "",
    "EditPassword:=" , "",
    "PasswordType:=" , "UnknownPassword",
    "HideContents:=" , True,
    "ReplaceNames:=" , True,
    "ComponentOutline:=" , "None",
    "IncludedParts:=" , ["Box1"],
    "HiddenParts:=" , [],
    "IncludedCS:=" , [],
    "DefaultHandle:=" , "Global",
    "IncludedParameters:=" , [],
    "IncludedDependentParameters:=", [],
    "ParameterDescription:=", [],
    "IsLicensed:=" , False,
    "LicensingDllName:=" , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "VendorComponentIdentifier:=", "",
    "PublicKeyFile:=" , ""
  ],
  [
    "NAME:DesignData",
    "Excitations:=" , ["CoilTerminal1","CoilTerminal2","Winding1","External
    Circuit”]
  ],
  "F:/Designs/ComponentExternalCircuit/test.a3dcomp",
  [
    "NAME:ImageFile",
    "ImageFile:=" , ""
  ])
```

CreateBondwire

Creates a bondwire.

UI Access	**Draw** > **Bondwire**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:BondwireParameters",`
			`"WireType:=", <string("JEDEC_4Points", "JEDEC_5Points", or "LOW")>,`
			`"WireDiameter:=", <string>,`
			`"NumSides:=", <value>,`
			`"XPadPos:=", <value>,`
			`"YPadPos:=", <value>,`
			`"ZPadPos:=", <value>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"XDir:=", <value>, "YDir:=", <value>, "ZDir:=", <value>, "Distance:=", <value>, "h1:=", <value>, "h2:=", <value>, "alpha:=", <value>, "beta:=", <value>, "WhichAxis:=", <string("X","Y", or "Z")>, "ReverseDirection:=", <boolean>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateBondwire(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateBondwire(["NAME:BondwireParameters", "WireType:=" , "JEDEC_4Points", "WireDiameter:=" , "0.025mm", "NumSides:=" , "6", "XPadPos:=" , "1.6mm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "YPadPos:="             , "-0.2mm",
  "ZPadPos:="             , "0mm",
  "XDir:="                , "-2.2mm",
  "YDir:="                , "-1.4mm",
  "ZDir:="                , "0mm",
  "Distance:="            , "2.60768096208106mm",
  "h1:="                  , "0.2mm",
  "h2:="                  , "0mm",
  "alpha:="               , "80deg",
  "beta:="                , "0",
  "WhichAxis:="           , "Z",
  "ReverseDirection:="    , True
  ],
["NAME:Attributes",
  "Name:="                , "Bondwire1",
  "Flags:="               , "",
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"vacuum\"",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , True,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateBox

Creates a box.

UI Access	**Draw** > **Box**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:BoxParameters",`
			`  "XPosition:=", <string>,`
			`  "YPosition:=", <string>,`
			`  "ZPosition:=", <string>,`
			`  "XSize:="    , <string>,`
			`  "YSize:="    , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"ZSize:=" , <string>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateBox(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateBox(["NAME:BoxParameters", "XPosition:=" , "0.5mm", "YPosition:=" , "-6.5mm", "ZPosition:=" , "0mm", "XSize:=" , "2mm", "YSize:=" , "1.5mm", "ZSize:=" , "1.5mm"], ["NAME:Attributes", "Name:=" , "Box3", "Flags:=" , "", "Color:=" , "(143 175 143)", "Transparency:=" , 0, "PartCoordinateSystem:=" , "Global", "UDMId:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "MaterialValue:="        , "\"copper\"",
  "SurfaceMaterialValue:=" , "\"\"",
  "SolveInside:="          , False,
  "ShellElement:="         , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="   , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="        , False
])
```

CreateCircle

Creates a circle.

UI Access	**Draw** > **Circle**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:CircleParameters",`
			`  "IsCovered:=", <boolean>,`
			`  "XCenter:=", <value>,`
			`  "YCenter:=", <value>,`
			`  "ZCenter:=", <value>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Radius:=", <value>, "WhichAxis:=", <string> "NumSegments:=", <string containing integer>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateCircle(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateCircle(["NAME:CircleParameters", "IsCovered:=" , True, "XCenter:=" , "5.5mm", "YCenter:=" , "-3mm", "ZCenter:=" , "0mm", "Radius:=" , "0.707106781186548mm", "WhichAxis:=" , "Z", "NumSegments:=" , "0"], ["NAME:Attributes", "Name:=" , "Circle1", "Flags:=" , "", "Color:=" , "(143 175 143)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateCone

Creates a cone.

UI Access	**Draw** > **Cone**.		
Parameters	Name	Type	Description

	<*Parameters*>	Array	Structured array.
			`["NAME:ConeParameters",`
			`  "XCenter:=", <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"YCenter:=", <string>, "ZCenter:=", <string>, "WhichAxis:=", <string>, "Height:=", <string>, "BottomRadius:=", <string>, "TopRadius:=", <string>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateCone(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateCone(["NAME:ConeParameters", "XCenter:=" , "3mm", "YCenter:=" , "-4.5mm", "ZCenter:=" , "0mm", "WhichAxis:=" , "Z", "Height:=" , "2.5mm", "BottomRadius:=" , "2.82842712474619mm", "TopRadius:=" , "2.23606797749979mm"], ["NAME:Attributes",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Name:="                , "Cone1",
  "Flags:="               , "",
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateCutplane

Creates a cutplane.

UI Access	**Draw** > **Plane**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:PlaneParameters",
  "PlaneBaseX:=", <string>,
  "PlaneBaseY:=", <string>,
  "PlaneBaseZ:=", <string>,
  "PlaneNormalX:=", <string>,
  "PlaneNormalY:=", <string>,
  "PlaneNormalZ:=", <string>]</pre></td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>See: AttributesArray. CreateCutplane only takes the Name and Color attributes.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreateCutplane(<*Parameters*>, <*Attributes*>)

Python Example	<pre>oEditor.CreateCutplane(
	["NAME:PlaneParameters",
	  "PlaneBaseX:=" , "-0.6mm",
	  "PlaneBaseY:=" , "-0.8mm",
	  "PlaneBaseZ:=" , "0mm",
	  "PlaneNormalX:=" , "1.2mm",
	  "PlaneNormalY:=" , "0.2mm",
	  "PlaneNormalZ:=" , "0mm"</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	], ["NAME:Attributes", "Name:=" , "Plane1", "Color:=" , "(143 175 143)"])

CreateCylinder

Creates a cylinder.

UI Access	**Draw** > **Cylinder**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:CylinderParameters", "XCenter:=", <string>, "YCenter:=", <string>, "ZCenter:=", <string>, "Radius:=", <string>, "Height:=", <string>, "WhichAxis:=", <string>, "NumSides:=", <string containing integer>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	CreateCylinder(<*Parameters*>, <*Attributes*>)

Python Example	oEditor.CreateCylinder(
	["NAME:CylinderParameters",
	  "XCenter:=" , "6mm",
	  "YCenter:=" , "-4.5mm",
	  "ZCenter:=" , "0mm",
	  "Radius:=" , "0.5mm",
	  "Height:=" , "4.5mm",
	  "WhichAxis:=" , "Z",
	  "NumSides:=" , "0"
	],
	["NAME:Attributes",
	  "Name:=" , "Cylinder1",
	  "Flags:=" , "",
	  "Color:=" , "(143 175 143)",
	  "Transparency:=" , 0,
	  "PartCoordinateSystem:=", "Global",
	  "UDMId:=" , "",
	  "MaterialValue:=" , "\"copper\"",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateEllipse

Creates an ellipse.

UI Access	**Draw** > **Ellipse**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:EllipseParameters",`
			`"IsCovered:=", <string>,`
			`"XCenter:=", <string>,`
			`"YCenter:=", <string>,`
			`"ZCenter:=", <string>,`
			`"MajRadius:=", <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Ratio:=", <string>, "WhichAxis:=", <string>, "NumSegments:=", <string>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateEllipse(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateEllipse(["NAME:EllipseParameters", "IsCovered:=" , True, "XCenter:=" , "0.6mm", "YCenter:=" , "-0.6mm", "ZCenter:=" , "0mm", "MajRadius:=" , "0.2mm", "Ratio:=" , "7", "WhichAxis:=" , "Z", "NumSegments:=" , "0"], ["NAME:Attributes", "Name:=" , "Ellipse1", "Flags:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateEquationCurve

Creates an equation-based curve.

UI Access	**Draw** > **Equation-Based Curve**.		
Parameters	Name	Type	Description
	<*Parameters*>	Array	Structured array. ["NAME:EquationBasedCurveParameters",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"XtFunction:=", <string>, "YtFunction:=", <string>, "ZtFunction:=", <string>, "tStart:=", <string>, "tEnd:=", <string>, "NumOfPointsOnCurve:=", <string>, "Version:=", <integer>), <polylineArray(optional)>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateEquationCurve(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateEquationCurve(["NAME:EquationBasedCurveParameters", "XtFunction:=" , "1", "YtFunction:=" , "3", "ZtFunction:=" , "32", "tStart:=" , "1", "tEnd:=" , "3", "NumOfPointsOnCurve:=" , "0", "Version:=" , 1,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ["NAME:PolylineXSection",
    "XSectionType:="        , "None",
    "XSectionOrient:="      , "Auto",
    "XSectionWidth:="       , "0",
    "XSectionTopWidth:="    , "0",
    "XSectionHeight:="      , "0",
    "XSectionNumSegments:=" , "0",
    "XSectionBendType:="    , "Corner"
  ]
],
["NAME:Attributes",
  "Name:="                , "EquationCurve1",
  "Flags:="               , "",
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateEquationSurface

Creates an equation-based surface.

UI Access: **Draw** > **Equation-Based Surface**.

Parameters

Name	Type	Description
<Parameters>	Array	Structured array.

```
["NAME:EquationBasedSurfaceParameters",
  "XuvFunction:=" , <string equation containing Function, Operators and/or quantities _u, _v, or PI>,
  "YuvFunction:=" , <string equation containing Function, Operators and/or quantities _u, _v, or PI>,
  "ZuvFunction:=" , <string equation containing Function, Operators and/or quantities _u, _v, or PI>,
  "uStart:=" , <string>,
  "uEnd:=" , <string>,
  "vStart:=" , <string>,
  "vEnd:=" , <string>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Version:=" , <integer>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateEquationSurface(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateEquationSurface(["NAME:EquationBasedSurfaceParameters", "XuvFunction:=" , "_u", "YuvFunction:=" , "_v", "ZuvFunction:=" , "sin(_u)+cos(_v)", "uStart:=" , "1", "uEnd:=" , "10", "vStart:=" , "1", "vEnd:=" , "10", "Version:=" , 1], ["NAME:Attributes", "Name:=" , "EquationSurface1", "Flags:=" , "", "Color:=" , "(143 175 143)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateHelix

Creates a helix based on a sweep of specified objects.

UI Access	**Draw** > **Helix**.

Parameters

Name	Type	Description
<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

<*Parameters*>	Array	Structured array.
		`["NAME:HelixParameters",`
		`  "XCenter:=" , <string>,`
		`  "YCenter:=" , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"ZCenter:=" , <string>, "XStartDir:=" , <string>, "YStartDir:=" , <string>, "ZStartDir:=" , <string>, "NumThread:=" , <string>, "RightHand:=" , <boolean>, "RadiusIncrement:=" , <string>, "Thread:=" , <string>]
Return Value	None.		

Python Syntax	CreateHelix(<*SelectionsArray*>, <*Parameters*>)
Python Example	oEditor.CreateHelix(["NAME:Selections", "Selections:=" , "EquationSurface2", "NewPartsModelFlag:=" , "Model"], ["NAME:HelixParameters", "XCenter:=" , "10000mm", "YCenter:=" , "40000mm", "ZCenter:=" , "0mm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "XStartDir:="            , "0mm",
  "YStartDir:="            , "10000mm",
  "ZStartDir:="            , "0mm",
  "NumThread:="            , "1",
  "RightHand:="            , True,
  "RadiusIncrement:="      , "0mm",
  "Thread:="               , "1mm"
])
```

CreatePoint

Creates a point.

UI Access	**Draw** > **Point**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:PointParameters",`
			`"PointX:=", <value>,`
			`"PointY:=", <value>,`
			`"PointZ:=", <value>]`

	<AttributesArray>	Array	Structured array. See: AttributesArray. CreatePoint takes only the Name and Color attributes.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreatePoint(<*Parameters*>, <*Attributes*>)

Python Example	`oEditor.CreatePoint(`
	`["NAME:PointParameters",`
	`  "PointX:=" , "0.2mm",`
	`  "PointY:=" , "-0.2mm",`
	`  "PointZ:=" , "0mm"`
	`  ],`
	`["NAME:Attributes",`
	`  "Name:="  , "Point1",`
	`  "Color:=" , "(143 175 143)"`
	`])`

CreatePolyline

Creates a polyline.

UI Access	**Draw** > **Line**.		
Parameters	Name	Type	Description

	<*Parameters*>	Array	Structured array.
			`["NAME:PolylineParameters",`
			`  "IsPolylineCovered:=", <bool>,`
			`  "IsPolylineClosed:=", <bool>,`
			`  <PolylinePointsArray>,`
			`  <PolylineSegmentsArray>]`

	<*PolylinePointsArray*>	Array	`["NAME:PolylinePoints", <OnePointArray>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<OnePointArray>, ...]
	<*OnePointArray*>	Array	["NAME:PLPoint", "X:=", <value>, "Y:=", <value>, "Z:=", <value>))
	<*PolylineSegmentsArray*>	Array	["NAME:PolylineSegments", <OneSegmentArray>, <OneSegmentArray>, ...]
	<*OneSegmentArray*>	Array	["NAME:PLSegment", "SegmentType:=", <"Line", "Arc", "Spline", or "AngularArc">, "StartIndex:=", <value>, "NoOfPoints:=", <value>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreatePolyline(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreatePolyline(["NAME:PolylineParameters", "IsPolylineCovered:=" , True, "IsPolylineClosed:=" , False, ["NAME:PolylinePoints", ["NAME:PLPoint", "X:=" , "20000mm", "Y:=" , "-20000mm", "Z:=" , "0mm"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    ["NAME:PLPoint",
    "X:="                   , "-90000mm",
    "Y:="                   , "20000mm",
    "Z:="                   , "0mm"
    ],
    ["NAME:PLPoint",
    "X:="                   , "10000mm",
    "Y:="                   , "-140000mm",
    "Z:="                   , "0mm"
    ]
  ],
  ["NAME:PolylineSegments",
    ["NAME:PLSegment",
    "SegmentType:="         , "Line",
    "StartIndex:="          , 0,
    "NoOfPoints:="          , 2
    ],
    ["NAME:PLSegment",
    "SegmentType:="         , "Line",
    "StartIndex:="          , 1,
    "NoOfPoints:="          , 2
    ]
  ],
  ["NAME:PolylineXSection",
    "XSectionType:="        , "None",
    "XSectionOrient:="      , "Auto",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"XSectionWidth:=" , "0mm", "XSectionTopWidth:=" , "0mm", "XSectionHeight:=" , "0mm", "XSectionNumSegments:=" , "0", "XSectionBendType:=" , "Corner"]], ["NAME:Attributes", "Name:=" , "Polyline1", "Flags:=" , "", "Color:=" , "(143 175 143)", "Transparency:=" , 0, "PartCoordinateSystem:=", "Global", "UDMId:=" , "", "MaterialValue:=" , "\"copper\"", "SurfaceMaterialValue:=", "\"\"", "SolveInside:=" , False, "ShellElement:=" , False, "ShellElementThickness:=", "0mm", "IsMaterialEditable:=" , True, "UseMaterialAppearance:=", False, "IsLightweight:=" , False])

CreateRectangle

Creates a rectangle.

UI Access	**Draw** > **Rectangle**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:RectangleParameters",`
		`"IsCovered:=" , <boolean>,`
		`"XStart:=" , <string>,`
		`"YStart:=" , <string>,`
		`"ZStart:=" , <string>,`
		`"Width:=" , <string>,`
		`"Height:=" , <string>,`
		`"WhichAxis:=" , <string "X", "Y", or "Z">]`

<AttributesArray>	Array	Structured array. See: AttributesArray.

Return Value

None.

Python Syntax	CreateRectangle(*<Parameters>*, *<Attributes>*)

Python Example	`oEditor.CreateRectangle(`
	`["NAME:RectangleParameters",`
	`"IsCovered:="  , True,`
	`"XStart:="  , "-80000mm",`
	`"YStart:="  , "-90000mm",`
	`"ZStart:="  , "0mm",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Width:="               , "20000mm",
  "Height:="              , "30000mm",
  "WhichAxis:="           , "Z"
],
["NAME:Attributes",
  "Name:="                , "Rectangle1",
  "Flags:="               , "",
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateRegion

Creates a region containing the design.

<table>
<tr><td>UI Access</td><td>Draw > Region.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:RegionParameters",

  "+XPaddingType:=", <string "Percentage Offset",
  "Absolute Offset", or "Absolute Position">,

  "+XPadding:=", <string X value>,

  "-XPaddingType:=", <string "Percentage Offset",
  "Absolute Offset", or "Absolute Position">,

  "-XPadding:=", <string -X value>,

  "+YPaddingType:=", <string "Percentage Offset",
  "Absolute Offset", or "Absolute Position">,

  "+YPadding:=", <string Y value>,

  "-YPaddingType:=", <string "Percentage Offset",
  "Absolute Offset", or "Absolute Position">,

  "-YPadding:=", <string -Y value>,

  "+ZPaddingType:=", <string "Percentage Offset",
  "Absolute Offset", or "Absolute Position">,

  "+ZPadding:=", <string Z value>,

  "-ZPaddingType:=", <string "Percentage Offset",</pre>
</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Absolute Offset", or "Absolute Position">, "-ZPadding:=", <string -Z value>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateRegion(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateRegion(["NAME:RegionParameters", "+XPaddingType:=" , "Percentage Offset", "+XPadding:=" , "0", "-XPaddingType:=" , "Percentage Offset", "-XPadding:=" , "0", "+YPaddingType:=" , "Percentage Offset", "+YPadding:=" , "0", "-YPaddingType:=" , "Percentage Offset", "-YPadding:=" , "0", "+ZPaddingType:=" , "Percentage Offset", "+ZPadding:=" , "0", "-ZPaddingType:=" , "Percentage Offset", "-ZPadding:=" , "0"], ["NAME:Attributes", "Name:=" , "Region", "Flags:=" , "Wireframe#", "Color:=" , "(143 175 143)", "Transparency:=" , 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"vacuum\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "nan ",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateRegularPolygon

Creates a regular polygon.

UI Access	**Draw** > **Regular Polygon**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:RegularPolygonParameters",`
			`  "IsCovered:=" , <boolean>,`
			`  "XCenter:=" , <string>,`
			`  "YCenter:=" , <string>,`
			`  "ZCenter:=" , <string>,`
			`  "XStart:=" , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"YStart:=" , <string>, "ZStart:=" , <string>, "NumSides:=" , <string containing number greater than 2>, "WhichAxis:=" , <string "X", "Y", or "Z"]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateRegularPolygon(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateRegularPolygon(["NAME:RegularPolygonParameters", "IsCovered:=" , True, "XCenter:=" , "-70000mm", "YCenter:=" , "-100000mm", "ZCenter:=" , "0mm", "XStart:=" , "-50000mm", "YStart:=" , "-80000mm", "ZStart:=" , "0mm", "NumSides:=" , "12", "WhichAxis:=" , "Z"], ["NAME:Attributes", "Name:=" , "Polygon1", "Flags:=" , "", "Color:=" , "(143 175 143)", "Transparency:=" , 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateRegularPolyhedron

Creates a regular polyhedron.

UI Access	**Draw** > **Regular Polyhedron**.

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:PolyhedronParameters",`
		`  "XCenter:=" , <string>,`
		`  "YCenter:=" , <string>,`
		`  "ZCenter:=" , <string>,`
		`  "XStart:=" , <string>,`
		`  "YStart:=" , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"ZStart:=" , <string>, "Height:=" , <string>, "NumSides:=" , <string containing number greater than 2>, "WhichAxis:=" , <string "X", "Y", or "Z">]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateRegularPolyhedron(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateRegularPolyhedron(["NAME:PolyhedronParameters", "XCenter:=" , "40000mm", "YCenter:=" , "-80000mm", "ZCenter:=" , "0mm", "XStart:=" , "50000mm", "YStart:=" , "-70000mm", "ZStart:=" , "0mm", "Height:=" , "50000mm", "NumSides:=" , "8", "WhichAxis:=" , "Z"], ["NAME:Attributes", "Name:=" , "RegularPolyhedron1", "Flags:=" , "", "Color:=" , "(143 175 143)", "Transparency:=" , 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:", False,
  "IsLightweight:="       , False
])
```

CreateSphere

Creates a sphere.

UI Access	**Draw** > **Sphere**.		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:SphereParameters",`
			`"XCenter:=", <string>,`
			`"YCenter:=", <string>,`
			`"ZCenter:=", <string>,`
			`"Radius:=", <string>]`

	<AttributesArray>	Array	Structured array. See: AttributesArray.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	CreateSphere(<*Parameters*>, <*Attributes*>)

Python Example	`oEditor.CreateSphere(`
	`["NAME:SphereParameters",`
	`  "XCenter:="  , "-40000mm",`
	`  "YCenter:="  , "-130000mm",`
	`  "ZCenter:="  , "0mm",`
	`  "Radius:="  , "22360.6797749979mm"`
	`],`
	`["NAME:Attributes",`
	`  "Name:="  , "Sphere1",`
	`  "Flags:="  , "",`
	`  "Color:="  , "(143 175 143)",`
	`  "Transparency:="  , 0,`
	`  "PartCoordinateSystem:=", "Global",`
	`  "UDMId:="  , "",`
	`  "MaterialValue:="  , "\"copper\"",`
	`  "SurfaceMaterialValue:=", "\"\"",`
	`  "SolveInside:="  , False,`
	`  "ShellElement:="  , False,`
	`  "ShellElementThickness:=", "0mm",`
	`  "IsMaterialEditable:="  , True,`
	`  "UseMaterialAppearance:=", False,`
	`  "IsLightweight:="  , False`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateSpiral

Creates a spiral by sweeping the specified object(s).

<table>
<tr><td>UI Access</td><td>Draw > Spiral.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:SpiralParameters",
  "XCenter:=" , <string>,
  "YCenter:=" , <string>,
  "ZCenter:=" , <string>,
  "YStartDir:=" , <string>,
  "ZStartDir:=" , <string>,
  "NumThread:=" , <string>,
  "RightHand:=" , <boolean>,
  "RadiusIncrement:=" , <string>]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreateSpiral(<*Parameters*>, <*Attributes*>)
Python Example	`oEditor.CreateSpiral(` `["NAME:Selections",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Selections:="          , "Polygon2",
  "NewPartsModelFlag:="   , "Model"
],
["NAME:SpiralParameters",
  "XCenter:="             , "-70000mm",
  "YCenter:="             , "50000mm",
  "ZCenter:="             , "0mm",
  "XStartDir:="           , "-60000mm",
  "YStartDir:="           , "-10000mm",
  "ZStartDir:="           , "0mm",
  "NumThread:="           , "1",
  "RightHand:="           , True,
  "RadiusIncrement:="     , "1mm"
])
```

CreateTorus

Creates a torus.

UI Access	**Draw** > **Torus**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. `["NAME:TorusParameters",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"XCenter:=" , <string>, "YCenter:=" , <string>, "ZCenter:=" , <string>, "MajorRadius:=" , <string>, "MinorRadius:=" , <string>, "WhichAxis:=" , <string "X", "Y", or "Z">]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateTorus(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateTorus(["NAME:TorusParameters", "XCenter:=" , "0.6mm", "YCenter:=" , "-0.6mm", "ZCenter:=" , "0mm", "MajorRadius:=" , "0.365028153987289mm", "MinorRadius:=" , "0.0821854415126694mm", "WhichAxis:=" , "Z"], ["NAME:Attributes", "Name:=" , "Torus1", "Flags:=" , "", "Color:=" , "(143 175 143)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , False,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:="       , False
])
```

CreateUserDefinedPart

Creates a user-defined part.

UI Access	**Draw** > **User-Defined Primitive** > [**Part**]		
Parameters	Name	Type	Description

	<Parameters>	Array	Structured array.
			`["NAME:UserDefinedPrimitiveParameters",`
			`"DllName:=" , <string>,`
			`"Version:=" , <string>,`
			`"NoOfParameters:=" , <integer>,`
			`"Library:=" , <string>,`
			`<paramVectorArray>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<paramVectorArray>	Array	Structured array containing arrays for each pair:
			`["NAME:ParamVector",`
			`  <pair>, <pair>, <pair>,...`
			`]`
			` <pair>:`
			`["NAME:Pair",`
			`  "Name:=" , <string>,`
			`  "Value:=" , <string>]`

	<Attributes>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateUserDefinedPart(*<Parameters>*, *<paramVectorArray>*, *<Attributes>*)

Python Example	`oEditor.CreateUserDefinedPart(`
	`["NAME:UserDefinedPrimitiveParameters",`
	`"DllName:=" , "RMxprt/LapCoil.dll",`
	`"Version:=" , "16.0",`
	`"NoOfParameters:=" , 22,`
	`"Library:=" , "syslib",`
	`  ["NAME:ParamVector",`
	`  ["NAME:Pair",`
	`  "Name:=" , "DiaGap",`
	`  "Value:=" , "100mm"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    ["NAME:Pair",
    "Name:=" , "DiaYoke",
    "Value:=" , "20mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Length",
    "Value:=" , "100mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Skew",
    "Value:=" , "0deg"
    ],
    ["NAME:Pair",
    "Name:=" , "Slots",
    "Value:=" , "18"
    ],
    ["NAME:Pair",
    "Name:=" , "SlotType",
    "Value:=" , "1"
    ],
    ["NAME:Pair",
    "Name:=" , "Hs0",
    "Value:=" , "1mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Hs1",
    "Value:=" , "1mm"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    ["NAME:Pair",
    "Name:=" , "Hs2",
    "Value:=" , "10mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Bs0",
    "Value:=" , "2.5mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Bs1",
    "Value:=" , "8mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Bs2",
    "Value:=" , "5mm"
    ],
    ["NAME:Pair",
    "Name:=" , "Rs",
    "Value:=" , "0mm"
    ],
    ["NAME:Pair",
    "Name:=" , "FilletType",
    "Value:=" , "0"
    ],
    ["NAME:Pair",
    "Name:=" , "Layers",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Value:=" , "2"
    ],
    ["NAME:Pair",
    "Name:=" , "CoilPitch",
    "Value:=" , "4"
    ],
    ["NAME:Pair",
    "Name:=" , "EndExt",
    "Value:=" , "5mm"
    ],
    ["NAME:Pair",
    "Name:=" , "SpanExt",
    "Value:=" , "25mm"
    ],
    ["NAME:Pair",
    "Name:=" , "BendAngle",
    "Value:=" , "0deg"
    ],
    ["NAME:Pair",
    "Name:=" , "SegAngle",
    "Value:=" , "10deg"
    ],
    ["NAME:Pair",
    "Name:=" , "LenRegion",
    "Value:=" , "200mm"
    ],
    ["NAME:Pair",
    "Name:=" , "InfoCoil",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Value:=" , "0"
    ]
  ]
],
["NAME:Attributes",
  "Name:=" , "LapCoil1",
  "Flags:=" , "",
  "Color:=" , "(143 175 143)",
  "Transparency:=" , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:=" , "",
  "MaterialValue:=" , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:=" , False,
  "ShellElement:=" , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:=" , True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:=" , False
])
```

Edit3DComponent

Edits a specified 3D component by saving it to a new name so that its properties are accessible. To change the properties on an existing component without changing its name, see ChangeProperty.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*compName*>	String	Component name.

<*Parameters*>	Array	Structured array.
		`["NAME:EditComponentParametersData",`
		`"NewComponentName:=", <string>,`
		`"GeometryParameters:=", <string>,`
		`"MaterialParameters:=", <string>,`
		`"DesignParameters:=", <string>,`
		`<ComponentMeshing>,`
		`<Excitations>]`

<*ComponentMeshing*>	Array	Structured array.
		`["NAME:Component Meshing",`
		`"MeshAssembly:=", <boolean>]`

<*Excitations*>	Array	Structured array containing array of suppressed excitations.
		`["NAME:Excitations",`
		`"Suppressed:=", <array>]`

Return Value

None.

Python Syntax	Edit3DComponent (<*compName*>, <*Parameters*>)

Python Example	`oEditor.Edit3DComponent(`
	`"Connector1",`
	`["NAME:EditComponentParametersData",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NewComponentName:=", "Connector2",
    "GeometryParameters:=", "",
    "MaterialParameters:=", "",
    "DesignParameters:=", "",
  ["NAME:Component Meshing",
    "MeshAssembly:=", False],
  ["NAME:Excitations",
    "Suppressed:=", []]
  ]
)
```

Edit3DComponentDefinition

Edits definitions of a specified 3D component.

UI Access	N/A

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:EditComponentParametersData",`
		`  "OriginalComponentName:=", <string>,`
		`  "NewComponentName:=", <string>,`
		`  "GeometryParameters:=", <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"MaterialParameters:=", <string>, "DesignParameters:=", <string>, <ComponentMeshing>, <Excitations>]
	<ComponentMeshing>	Array	Structured array. ["NAME:Component Meshing", "MeshAssembly:=", <boolean>]
	<Excitations>	Array	Structured array containing array of suppressed excitations. ["NAME:Excitations", "Suppressed:=", <array>]
Return Value	None.		
Python Syntax	Edit3DComponent (*<Parameters>*)		
Python Example	oEditor.Edit3DComponent(["NAME:EditComponentParametersData", "OriginalComponentName:=", "Connector1", "NewComponentName:=", "Connector2", "GeometryParameters:=", "", "MaterialParameters:=", "", "DesignParameters:=", "", ["NAME:Component Meshing", "MeshAssembly:=", False],		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:Excitations", "Suppressed:=", []]])

EditPolyline

Modifies a specified polyline. See: CreatePolyline.

UI Access	N/A		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.
	<Parameters>	Array	Structured array. ["NAME:PolylineParameters", "IsPolylineCovered:=", <bool>, "IsPolylineClosed:=", <bool>, <PolylinePointsArray>, <PolylineSegmentsArray>]
	<PolylinePointsArray>	Array	["NAME:PolylinePoints", <OnePointArray>, <OnePointArray>, ...]
	<OnePointArray>	Array	["NAME:PLPoint", "X:=", <value>, "Y:=", <value>, "Z:=", <value>]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<PolylineSegmentsArray>	Array	<PolylineSegmentsArray> ["NAME:PolylineSegments", <OneSegmentArray>, <OneSegmentArray>, ...]
	<OneSegmentArray>	Array	["NAME:PLSegment", "SegmentType:=", <"Line", "Arc", "Spline", or "AngularArc">, "StartIndex:=", <value>, "NoOfPoints:=", <value>]
Return Value	None.		

Python Syntax	EditPolyline(<*SelectionsArray*>, <*Parameters*>)
Python Example	oEditor.EditPolyline(["NAME:Selections", "Selections:=", "Polyline1"] ["NAME:PolylineParameters", "IsPolylineCovered:=" , True, "IsPolylineClosed:=" , False, ["NAME:PolylinePoints", ["NAME:PLPoint", "X:=" , "20000mm", "Y:=" , "-20000mm", "Z:=" , "0mm"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    ["NAME:PLPoint",
      "X:="                    , "-90000mm",
      "Y:="                    , "20000mm",
      "Z:="                    , "0mm"
    ],
    ["NAME:PLPoint",
      "X:="                    , "10000mm",
      "Y:="                    , "-140000mm",
      "Z:="                    , "0mm"
    ]
  ],
  ["NAME:PolylineSegments",
    ["NAME:PLSegment",
    "SegmentType:="          , "Line",
    "StartIndex:="           , 0,
    "NoOfPoints:="           , 2
    ],
    ["NAME:PLSegment",
    "SegmentType:="          , "Line",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "StartIndex:="          , 1,
  "NoOfPoints:="          , 2
  ]
],
["NAME:PolylineXSection",
  "XSectionType:="        , "None",
  "XSectionOrient:="      , "Auto",
  "XSectionWidth:="       , "0mm",
  "XSectionTopWidth:="    , "0mm",
  "XSectionHeight:="      , "0mm",
  "XSectionNumSegments:=" , "0",
  "XSectionBendType:="    , "Corner"
]]
)
```

Get3DComponentDefinitionNames

Gets names of 3D component definitions.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing component definition names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Get3DComponentDefinitionNames()
Python Example	`oEditor.Get3DComponentDefinitionNames()`

Get3DComponentInstanceNames

Returns instance names of 3D component definitions.

UI Access	N/A
Parameters	Name | Type | Description <*DefinitionName*> | String | Definition name.
Return Value	Array containing instance names.

Python Syntax	Get3DComponentInstanceNames(<*DefinitionName*>)
Python Example	`oEditor.Get3DComponentInstanceNames("Connector")`

Get3DComponentMaterialNames

Returns material names for a specified *.a3dcomp format 3D component.

UI Access	N/A
Parameters	Name | Type | Description <*InstanceName*> | String | Component instance name.
Return Value	Array containing material names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Get3DComponentMaterialNames(<*InstanceName*>)
Python Example	`oEditor.Get3DComponentMaterialNames("Connector1.a3dcomp")`

Get3DComponentMaterialProperties

Returns material properties for a specified 3D component.

UI Access	N/A
Parameters	Name | Type | Description <*MaterialName*> | String | Material name.
Return Value	Array containing material properties.

Python Syntax	Get3DComponentMaterialProperties(<*MaterialName*>)
Python Example	`oEditor.Get3DComponentMaterialProperties('Connector1:Material01')`

Get3DComponentParameters

Returns parameters for a specified 3D component.

UI Access	N/A
Parameters	Name | Type | Description <*compName*> | String | 3D component name.
Return Value	Array containing component parameters.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Get3DComponentParameters(<*compName*>)
Python Example	`oEditor.Get3DComponentParameters('Connector')`

Get3DComponentPartNames

Returns part names for a specified 3D component.

UI Access	N/A
Parameters	Name | Type | Description <*InstanceName*> | String | 3D component instance.
Return Value	Array containing part names.

Python Syntax	Get3DComponentParameters(<*InstanceName*>)
Python Example	`oEditor.Get3DComponentPartNames('Connector')`

Insert3DComponent

Inserts a 3D component in specified coordinate system with specified offset.

UI Access	**Draw** > **3D Component Library** > **Browse** > [**Component**].
Parameters	Name | Type | Description <*ComponentData*> | Array | Structured array. `["NAME:InsertComponentData",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"TargetCS:=", <string>, "ComponentFile:=", <string filepath>), "IsLocal:", <Boolean>, [InstanceParametersArray], "XOffset:=" , "float with unit", "YOffset:=" , "float with unit", "ZOffset:=" , "float with unit"]
Return Value	None.		

Python Syntax	Insert3DComponent(<*ComponentData*>)
Python Example	oEditor.Insert3DComponent(["NAME:InsertComponentData", "TargetCS:=" , "Global", "ComponentFile:=" , "C:/Program Files/ANSYS Inc/v261/AnsysEM/syslib/3DComponents/HFSS/Rectangular Waveguide/Straight.a3dcomp", "IsLocal:=" , False, "UniqueIdentifier:=" , "", ["NAME:InstanceParameters", "GeometryParameters:=" , "Flange=\'0.4in\' FlangeFillet=\'0.2in\' FlangeThickness=\'0.2in\' GuideHeight=\'0.4in\' GuideLength=\'4in\'

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      GuideWidth=\'0.9in\' WallThickness=\'0.1in\'",
      "MaterialParameters:=" , "",
      "DesignParameters:=" , ""
    ],
    "XOffset:=" , "-70mm",
    "YOffset:=" , "45mm",
    "ZOffset:=" , "0mm"
  ])
```

InsertComponent

Inserts a component.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ComponentData*></td><td>Array</td><td>Structured array. ["NAME:InsertComponentData", "Parameters:=", <string>, "TargetCS:=", <string>, "ComponentFile:=", <string filepath>]</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	InsertComponent(<*ComponentData*>)

Python Example	`oEditor.InsertComponent(`
	`["NAME:InsertComponentData",`
	`"Parameters:=", "",`
	`"TargetCS:=", "Global",`
	`"ComponentFile:=", "C:\tmp\Connector.a3dcomp"])`

InsertPolylineSegment

Insets a polyline segment before or after a specified existing segment.

UI Access	**Draw** > **Line Segment** > [**Selection**].		
Parameters	Name	Type	Description

	<*Parameters*>	Array	Structured array.
			`["NAME:Insert Polyline Segment",`
			`"Selections:=" , <string>,`
			`"Segment Indices:=" , <array containing integers>,`
			`"At Start:=" , <boolean>,`
			`"SegmentType:=" , <string "Line", "Arc", "Spline", or "AngularArc">,`
			`<PolylinePointsArray>]`

	<*PolylinePointsArray*>	Array	Structured array. See: CreatePolyline.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	InsertPolylineSegment(<*Parameters*>)
Python Example	oEditor.InsertPolylineSegment(["NAME:Insert Polyline Segment", "Selections:=" , "Polyline1:CreatePolyline:1", "Segment Indices:=" , [0], "At Start:=" , True, "SegmentType:=" , "Line", ["NAME:PolylinePoints", ["NAME:PLPoint", "X:=" , "1.1mm", "Y:=" , "0.8mm", "Z:=" , "0mm"], ["NAME:PLPoint", "X:=" , "0.6mm", "Y:=" , "-0.8mm", "Z:=" , "0mm"]]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SweepAlongPath

Sweeps the specified 1D or 2D parts along a path. The last 1D object specified is the path for the sweep.

<table>
<tr><td>UI Access</td><td>Draw > Sweep > Along Path.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><PathSweepParametersArray></td><td>Array</td><td>["NAME:PathSweepParameters",
"DraftAngle:=", <value>,
"DraftType:=", <string>,
Possible values for DraftType are "Extended", "Round", and "Natural".
"CheckFaceFaceIntersection:=", <bool>,
"TwistAngle:=", <string value with unit>,
"LockDirectionX:=" , <string value,
"LockDirectionY:=" , <string value,
"LockDirectionZ:=" , <string value>
]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	SweepAlongPath(<*SelectionsArray*>, <*PathSweepParametersArray*>)
Python Example	`oEditor = oDesign.SetActiveEditor("3D Modeler")` `oEditor.SweepAlongPath(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  [
    "NAME:Selections",
    "Selections:=" , "Section_1_1,WindingPath_1_1_1",
    "NewPartsModelFlag:=" , "Model"
  ],
  [
    "NAME:PathSweepParameters",
    "DraftAngle:=" , "0deg",
    "DraftType:=" , "Round",
    "CheckFaceFaceIntersection:=", False,
    "ClearAllIDs:=" , False,
    "TwistAngle:=" , "0deg",
    "LockDirectionX:=" , "0",
    "LockDirectionY:=" , "0",
    "LockDirectionZ:=" , "1"
  ])
```

SweepAlongVector

Sweeps the specified 1D or 2D parts along a vector.

UI Access	**Draw** > **Sweep** > **Along Vector**.		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
	<*VecSweepParametersArray*>	Array	`["NAME:VectorSweepParameters",` `  "DraftAngle:=", <value>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"DraftType:=", <string>, "CheckFaceFaceIntersection:=", <bool>, "SweepVectorX:=", <value> "SweepVectorY:=", <value> "SweepVectorZ:=", <value>] Possible values for DraftType are "Extended", "Round", and "Natural".
Return Value	None.		

Python Syntax	SweepAlongVector(<SelectionsArray>, <VecSweepParametersArray>)
Python Example	oEditor.SweepAlongVector(["NAME:Selections", "Selections:=" , "Rectangle1", "NewPartsModelFlag:=" , "Model"], ["NAME:VectorSweepParameters", "DraftAngle:=" , "0deg", "DraftType:=" , "Round", "CheckFaceFaceIntersection:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"SweepVectorX:=" , "0mm" "SweepVectorY:=" , "0mm" "SweepVectorX:=" , "12mm"])

SweepAroundAxis

Sweeps the specified 1D or 2D parts around an axis.

UI Access	**Draw** > **Sweep** > **Around Axis**.		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.
	<AxisSweepParametersArray>	Array	["NAME:AxisSweepParameters", "DraftAngle:=", <value>, "DraftType:=", <string>, "CheckFaceFaceIntersection:=", <bool>, "SweepAxis:=", <value> "SweepAngle:=", <value> "NumOfSegments:=", <value] Possible values for DraftType are "Extended", "Round", and "Natural". Possible values for SweepAxis are "X", "Y", and "Z".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	SweepAroundAxis(<*SelectionsArray*>, <*AxisSweepParametersArray*>)

Python Example	`oEditor.SweepAroundAxis(`
	`  [`
	`  "NAME:Selections",`
	`  "Selections:=" , "Rectangle1",`
	`  "NewPartsModelFlag:=" , "Model"`
	`  ],`
	`  [`
	`  "NAME:AxisSweepParameters",`
	`  "DraftAngle:=" , "0deg",`
	`  "DraftType:=" , "Round",`
	`  "CheckFaceFaceIntersection:=", False,`
	`  "SweepAxis:=" , "X"`
	`  "SweepAngle:=" , "360deg"`
	`  "NumOfSegments:=" , "12"`
	`])

SweepFacesAlongNormal

Sweep the specified face(s) along normal.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Modeler > Surface > Sweep Faces Along Normal</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><parameters></td><td>Array</td><td>Structured array.
["NAME:Parameters",
"NAME:SweepFaceAlongNormalToParameters",
"FacesToDetach:=", <faceIDarray>,
"LengthOfSweep:=", "<value><units>"]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	SweepFacesAlongNormal(<*SelectionsArray*> <*parameters*>)
Python Example	oEditor.SweepFacesAlongNormal(["NAME:Selections", "Selections:=", "Rectangle1", "NewPartsModelFlag:=", "Model"], ["NAME:Parameters", "NAME:SweepFaceAlongNormalToParameters", "FacesToDetach:=", [183], "LengthOfSweep:=", "0.1mm"])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SweepFacesAlongNormalWithAttributes

Sweep a face along normal, and specify attributes of the new object.

<table>
<tr><td>UI Access</td><td colspan="3">Modeler > Surface > Sweep Faces Along Normal</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><parameters></td><td>Array</td><td>["NAME:Parameters",
"NAME:SweepFaceAlongNormalToParameters",
"FacesToDetach:=", <faceIDarray>,
"LengthOfSweep:=", "<value><units>"]</td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>Structured array. See: AttributesArray.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

Python Syntax	SweepFacesAlongNormalWithAttributes(<*SelectionsArray*>, <*parameters*>, <*AttributesArray*>)
Python Example	oEditor.SweepFacesAlongNormalWithAttributes(["NAME:Selections", "Selections:=", "Rectangle1", "NewPartsModelFlag:=", "Model"], ["NAME:Parameters", "NAME:SweepFaceAlongNormalToParameters", "FacesToDetach:=", [183],

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "LengthOfSweep:=", "0.1mm"],
["NAME:Attributes",
  "Name:=", "Box3",
  "Flags:=", "",
  "Color:=", "(143 175 143)",
  "Transparency:=", 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:=", "",
  "MaterialValue:=", "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:=", False,
  "ShellElement:=", False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:=", True,
  "UseMaterialAppearance:=", False,
  "IsLightweight:=", False])
```

UpdateComponentDefinition

Updates a 3D component's definition.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Draw** > **3D Component Library** > **Definitions**.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*data*></td><td>Array</td><td>Structured array.
	`["NAME:UpdateDefinitionData",`
	`"DefinitionNames:=", <string>,`
	`"Passwords:=", <array of strings>]`</td></tr></table>

Return Value	None.

Python Syntax	UpdateComponentDefinition(<*data*>)

Python Example	`oEditor.UpdateComponentDefinition(`
	`['NAME:UpdateDefinitionData',`
	`'DefinitionNames:=', 'Connector, Magic_Tee',`
	`'Passwords:=', ['','']`
	`]`
	`)`

Edit Menu Commands

Copy

DeletePolylinePoint

DuplicateAlongLine

DuplicateAroundAxis

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DuplicateMirror

Mirror

Move

OffsetFaces

Paste

Rotate

Scale

Copy

Copies specified part(s) to the clipboard.

UI Access	N/A
Parameters	Name | Type | Description <*SelectionsArray*> | Array | Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	Copy(<*SelectionsArray*>)

Python Example	`oEditor.Copy([`
	`  "NAME:Selections",`
	`  "Selections:=", "Box1"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeletePolylinePoint

Deletes either a start point or an end point from an existing polyline segment.

<table>
<tr><td>UI Access</td><td colspan="3">Edit > Delete [Start/End] Point</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><DeletePointArray></td><td>Array</td><td>Structured array.
["NAME:Delete Point",
"Selections:=", <string "<PolylineName>:<PolylineAction>:<int>">,
"Segment Index:=", <integer>,
"At Start:=", <bool True for start point; False for end point>]</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
</table>

Python Syntax	DeletePolylinePoint (<*DeletePointArray*>)
Python Example	oEditor.DeletePolylinePoint(["NAME:Delete Point", "Selections:=", "Polyline1:CreatePolyline:1", "Segment Index:=", 1, "At Start:=", True])

DuplicateAlongLine

Duplicates specified parts along a line.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Edit > Duplicate > Along Line.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array.
["NAME:Selections",
"Selections:=" , <string>,
"NewPartsModelFlag:=" , <string>]</td></tr>
<tr><td><ParametersArray></td><td>Array</td><td>Structured array.
["NAME:DuplicateToAlongLineParameters",
"CreateNewObjects:=" , <boolean>,
"XComponent:=" , <string>,
"YComponent:=" , <string>,
"ZComponent:=" , <string>,
"NumClones:=" , <string containing number greater than 1>]</td></tr>
<tr><td><OptionsArray></td><td>Array</td><td>Structured array.
["NAME:Options",
"DuplicateAssignments:=", <boolean>]</td></tr>
<tr><td><CreateGroup></td><td>Array</td><td>Optional. Structured array.
["CreateGroupsForNewObjects:=", <boolean]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DuplicateAlongLine (<*SelectionsArray*>, <*ParametersArray*>, <*OptionsArray*>, <*CreateGroup*>)
Python Example	oEditor.DuplicateAlongLine(["NAME:Selections", "Selections:=" , "Box1", "NewPartsModelFlag:=" , "Model"], ["NAME:DuplicateToAlongLineParameters", "CreateNewObjects:=" , False, "XComponent:=" , "1mm", "YComponent:=" , "-0.7mm", "ZComponent:=" , "0mm", "NumClones:=" , "2"], ["NAME:Options", "DuplicateAssignments:=", False], ["CreateGroupsForNewObjects:=", False])

DuplicateAroundAxis

Duplicates specified parts around an axis.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Edit > Duplicate > Around Axis.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array.
["NAME:Selections",
"Selections:=" , <string>,
"NewPartsModelFlag:=" , <string>]</td></tr>
<tr><td><ParametersArray></td><td>Array</td><td>Structured array.
["NAME:DuplicateAroundAxisParameters",
"CreateNewObjects:=" , <boolean>,
"WhichAxis:=" , <string>,
"AngleStr:=" , <string>,
"NumClones:=" , <string containing number greater than 1>]</td></tr>
<tr><td><OptionsArray></td><td>Array</td><td>Structured array.
["NAME:Options",
"DuplicateAssignments:=", <boolean>]</td></tr>
<tr><td><CreateGroup></td><td>Array</td><td>Optional. Structured array.
["CreateGroupsForNewObjects:=", <boolean>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DuplicateAroundAxis (<*SelectionsArray*>, <*ParametersArray*>, <*OptionsArray*>, <*CreateGroup*>)
Python Example	oEditor.DuplicateAroundAxis(["NAME:Selections", "Selections:=" , "Box1", "NewPartsModelFlag:=" , "Model"], ["NAME:DuplicateAroundAxisParameters", "CreateNewObjects:=" , True, "WhichAxis:=" , "Z", "AngleStr:=" , "90deg", "NumClones:=" , "2"], ["NAME:Options", "DuplicateAssignments:=", False], ["CreateGroupsForNewObjects:=", False])

DuplicateMirror

Duplicates specified parts according to a mirror plane.

UI Access	**Edit** > **Duplicate** > **Mirror**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description

<*SelectionsArray*>	Array	Structured array.
		`["NAME:Selections",`
		`"Selections:=" , <string>,`
		`"NewPartsModelFlag:=" , <string>]`

<*ParametersArray*>	Array	Structured array.
		`["NAME:DuplicateToMirrorParameters",`
		`"DuplicateMirrorBaseX:=", <string>,`
		`"DuplicateMirrorBaseY:=", <string>,`
		`"DuplicateMirrorNormalX:=", <string>,`
		`"DuplicateMirrorNormalY:=", <string>,`
		`"DuplicateMirrorNormalZ:=", <string>]`

<*OptionsArray*>	Array	Structured array.
		`["NAME:Options",`
		`"DuplicateAssignments:=", <boolean>]`

<*CreateGroup*>	Array	Optional. Structured array. `["CreateGroupsForNewObjects:=", <boolean>]`

Return Value

None.

Python Syntax	DuplicateMirror (<*SelectionsArray*>, <*ParametersArray*>, <*OptionsArray*>, <*CreateGroup*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oEditor.DuplicateMirror(["NAME:Selections", "Selections:=" , "Box1", "NewPartsModelFlag:=" , "Model"], ["NAME:DuplicateToMirrorParameters", "DuplicateMirrorBaseX:=", "-0.4mm", "DuplicateMirrorBaseY:=", "-1.2mm", "DuplicateMirrorBaseZ:=", "0mm", "DuplicateMirrorNormalX:=", "0.124034734589208mm", "DuplicateMirrorNormalY:=", "0.992277876713668mm", "DuplicateMirrorNormalZ:=", "0mm"], ["NAME:Options", "DuplicateAssignments:=", False], ["CreateGroupsForNewObjects:=", False])

Mirror

Mirrors specified part(s).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Edit > Arrange > Mirror.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><MirrorParameters></td><td>Array</td><td>Structured array.
<pre>["NAME:MirrorParameters",
  "MirrorBaseX:=" , <string>,
  "MirrorBaseY:=" , <string>,
  "MirrorBaseZ:=" , <string>,
  "MirrorNormalX:=" , <string>,
  "MirrorNormalY:=" , <string>,
  "MirrorNormalZ:=" , <string>]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	Mirror(<*SelectionsArray*>, <*MirrorParameters*>)

Python Example	<pre>oEditor.Mirror(
	["NAME:Selections",
	  "Selections:=" , "Box1_1",
	  "NewPartsModelFlag:=" , "Model"
	],
	["NAME:MirrorParameters",
	  "MirrorBaseX:=" , "-0.2mm",
	  "MirrorBaseY:=" , "-1.2mm",</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "MirrorBaseZ:="          , "0mm",
  "MirrorNormalX:="        , "-0.316227766016838mm",
  "MirrorNormalY:="        , "0.948683298050514mm",
  "MirrorNormalZ:="        , "0mm"
])
```

Move

Moves specified part(s).

UI Access	**Edit** > **Arrange** > **Move**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*TranslateParameters*></td><td>Array</td><td>Structured array. ["NAME:TranslateParameters", "TranslateVectorX:=" , <string>, "TranslateVectorY:=" , <string>, "TranslateVectorZ:=" , <string>]</td></tr></table>
Return Value	None.

Python Syntax	Move(<*SelectionsArray*>, <*TranslateParameters*>)
Python Example	oEditor.Move(["NAME:Selections", "Selections:=" , "Box1_1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "NewPartsModelFlag:="   , "Model"
],
["NAME:TranslateParameters",
  "TranslateVectorX:="    , "-0.5mm",
  "TranslateVectorY:="    , "0.1mm",
  "TranslateVectorZ:="    , "0mm"
])
```

OffsetFaces

Offsets the faces of selected part(s).

UI Access	**Edit** > **Arrange** > **Offset**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*OffsetParameters*></td><td>Array</td><td>Structured array. `["NAME:OffsetParameters",` `  "OffsetDistance:=" , <string>]`</td></tr></table>
Return Value	None.

Python Syntax	OffsetFaces (<*SelectionsArray*>, <*OffsetParameters*>)
Python Example	`oEditor.OffsetFaces(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:Selections",
	  "Selections:=", "Box1_1",
	  "NewPartsModelFlag:=", "Model"
	],
	["NAME:OffsetParameters",
	  "OffsetDistance:=", "16mm"
	])

Paste (Model Editor)

Pastes previously copied object(s). See: Copy.

UI Access	**Edit** > **Paste**.
Parameters	None.
Return Value	None.

Python Syntax	Paste()
Python Example	oEditor.Copy(["NAME:Selections", "Selections:=" , "Box1_2"]) oEditor.Paste()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Rotate

Rotates specified object(s).

<table>
<tr><td>UI Access</td><td>Edit > Arrange > Rotate.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><RotateParameters></td><td>Array</td><td>Structured array.
["NAME:RotateParameters",
"RotateAxis:=" , <string "X", "Y", or "Z">,
"RotateAngle:=" , <string>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	Rotate(<*SelectionsArray*>, <*RotateParameters*>)
Python Example	oEditor.Rotate(["NAME:Selections", "Selections:=", "Box1_1", "NewPartsModelFlag:=", "Model"], ["NAME:RotateParameters", "RotateAxis:=", "Z", "RotateAngle:=", "90deg"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Scale

Scales specified object(s).

UI Access	**Edit** > **Scale**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*ScaleParameters*></td><td>Array</td><td>Structured array. ["NAME:ScaleParameters", "ScaleX:=" , <string containing scale factor>, "ScaleY:=" , <string containing scale factor>, "ScaleZ:=" , <string containing scale factor>]</td></tr></table>
Return Value	None.

Python Syntax	Scale (<*SelectionsArray*>, <*ScaleParameters*>)
Python Example	oEditor.Scale(["NAME:Selections", "Selections:=", "Box1", "NewPartsModelFlag:=", "Model"], ["NAME:ScaleParameters",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"ScaleX:=", "2", "ScaleY:=", "2", "ScaleZ:=", "2"])

Modeler Menu Commands

AssignMaterial

Chamfer

Connect

CoverLines

CoverSurfaces

CreateEntityList

CreateFaceCS

CreateGroup

CreateObjectCS

CreateObjectFromEdges

CreateObjectFromFaces

CreateRelativeCS

DeleteEmptyGroups

DeleteLastOperation

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DetachFaces

EditEntityList

EditFaceCS

EditObjectCS

EditRelativeCS

Export

ExportModelImageToFile

ExportModelMeshToFile

Fillet

FlattenGroup

Generate History

HealObject

Import

ImportDXF

ImportGDSII [Modeler]

Intersect

MoveCStoEnd

MoveEntityToGroup

MoveFaces

ProjectSheet

PurgeHistory

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ReplaceWith3DComponent

Section

SeparateBody

SetModelUnits

SetWCS

ShowWindow

Split

Subtract

SweepFacesAlongNormal

ThickenSheet

UncoverFaces

Unite

Ungroup

WrapSheet

AlignFaces

Aligns the adjacent selected faces of imported objects which have only one operation in their History Tree.

UI Access	**Modeler** > **Model Preparation** > **Align Faces**		
Parameters	Name	Type	Description
	<*argFaceList*>	Array	["NAME:<SpecifiedName>", "BaseFaces:=" , <FaceNumberArray>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<table><tr><td></td><td></td><td>"SnapFaces:=" , <FaceNumberArray>]</td></tr><tr><td><*FaceNumberArray*></td><td>Array</td><td>Array of number represents specified faces.</td></tr></table>
Return Value	None.

Python Syntax	AlignFaces(<*argFaceList*>)

Python Example	`oEditor.AlignFaces([`
	`"NAME:Entity List",`
	`"BaseFaces:=" , [9],`
	`"SnapFaces:=" , [18]`
	`])`

AssignMaterial

Assigns a material to specified object(s).

UI Access	**Modeler** > **Assign Material**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*AttributesArray*></td><td>Array</td><td>Structured array. See: AttributesArray. This script supports the following attributes: - MaterialValue - SolveInside - ShellElement - ShellElementThickness </td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• IsMaterialEditable • UseMaterialAppearance • IsLightweight
Return Value	None.		

Python Syntax	AssignMaterial(<*SelectionsArray*>, <*AttributesArray*>)

Python Example	`oEditor.AssignMaterial(`
	`  ["NAME:Selections",`
	`  "AllowRegionDependentPartSelectionForPMLCreation:=", True,`
	`  "AllowRegionSelectionForPMLCreation:=", True,`
	`  "Selections:=" , "Box1"`
	`  ],`
	`  ["NAME:Attributes",`
	`  "MaterialValue:=" , "diamond",`
	`  "SolveInside:=" , False,`
	`  "ShellElement:=" , False,`
	`  "ShellElementThickness:=" , "nan",`
	`  "IsMaterialEditable:=" , True,`
	`  "UseMaterialAppearance:=" , False,`
	`  "IsLightweight:=" , False`
	`])`

Chamfer

Creates a chamfer.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Modeler > Chamfer.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:Parameters",
  ["NAME:ChamferParameters",
  "Edges:=" , <array containing integer>,
  "Vertices:=" , <array>,
  "LeftDistance:=" , <string>,
  "RightDistance:=" , <string>,
  "ChamferType:=" , <string "Symmetric",
  "Left Distance-Angle", "Right Distance-Angle", or
  "Left Distance-Right Distance">]
]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	Chamfer(<*SelectionsArray*>, <*Parameters*>)

Python Example	<pre>oEditor.Chamfer(
	["NAME:Selections",
	  "Selections:=" , "Box2",
	  "NewPartsModelFlag:=" , "Model"
	],
	["NAME:Parameters",
	  ["NAME:ChamferParameters",</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Edges:="               , [42],
    "Vertices:="            , [],
    "LeftDistance:="        , "0.1mm",
    "RightDistance:="       , "0.1mm",
    "ChamferType:="         , "Symmetric"
  ]
])
```

CleanUpModel

Cleans up history tree operations.

UI Access	**Modeler** > **Cleanup Model History**.
Parameters	None.
Return Value	None.

Python Syntax	CleanUpModel()
Python Example	`oEditor.CleanUpModel()`

Connect

Connects two or more 1D polyline objects or 2D sheet objects.

UI Access	**Modeler** > **Surface** > **Connect**.		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
Return Value	None.		

Python Syntax	Connect(<*SelectionsArray*>)

Python Example	`oEditor.Connect(`
	`  ["NAME:Selections",`
	`  "Selections:=", "Polyline2,Polyline1"`
	`  ])`

CoverLines

Covers two or more 1D objects to form a sheet.

UI Access	**Modeler** > **Surface** > **Cover Lines**.		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
Return Value	None.		

Python Syntax	CoverLines(<*SelectionsArray*>)

Python Example	`oEditor.CoverLines([`
	`  "NAME:Selections",`
	`  "Selections:=", "Polyline3,Polyline4",`
	`  "NewPartsModelFlag:=", "Model"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CoverSurfaces

Covers two or more faces to form a solid object.

UI Access	**Modeler** > **Surface** > **Cover Faces**.
Parameters	Name | Type | Description <*SelectionsArray*> | Array | Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	CoverSurfaces (<*SelectionsArray*>)

Python Example	`oEditor.CoverSurfaces(`
	`["NAME:Selections",`
	`  "Selections:=", "Obj1_Face1,Obj2_Face2",`
	`  "NewPartsModelFlag:=", "Model"`
	`])`

CreateEntityList

Creates a list of entities containing objects or faces (not both).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: As of the Ansys Electronics Desktop25r2 release, this command has been deprecated. It will be removed in a future release. Please use the CreateNamedSelection command instead.

<table>
<tr><td>UI Access</td><td colspan="3">Modeler > Named Selection > Create > [Object / Face] Selection</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:GeometryEntityListParameters",
"EntityType:=" , <string "Object" or "Face">,
"EntityList:=" , <string of object names or IDs>]
See GetObjectIDByName for returning object IDs.</td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>Structured array. See: AttributesArray. CreateEntityList takes only the "Name" parameter.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
</table>

Python Syntax	CreateEntityList(<*Parameters*>,<*AttributesArray*>)
Python Example	oEditor.CreateEntityList(["NAME:GeometryEntityListParameters", "EntityType:=", "Object", "EntityList:=", "Bondwire1,Bondwire2"], ["NAME:Attributes", "Name:=", "Objectlist1"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreateFaceCS

Creates a Face Coordinate System from a selected face.

UI Access	**Modeler** > **Coordinate System** > **Create** > **FaceCS**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:FaceCSParameters", <OriginArray>, "MoveToEnd:=", <boolean>, "FaceID:=", <integer>, <AxisPosnArray>, "WhichAxis:=" , <string "X", "Y", or "Z">, "ZRotationAngle:=" , <string>, "XOffset:=" , <string>, "YOffset:=" , <string>, "AutoAxis:=" , <boolean>]
	<OriginArray>	Array	Structured array. ["NAME:Origin", "IsAttachedToEntity:=" , <boolean>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "FaceCenter", "EdgeCenter", "OnVertex", "OnEdge", or "OnFace">, "UParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "VParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>] IsAttachedToEntity specifies whether the point is anchored to a vertex, edge, or face. If True, provide UParam and VParam. If False, provide XPosition, YPosition, and ZPosition to provide fixed position. Pass "0" for unused parameters.
	<AxisPosnArray>	Array	Structured array. ["NAME:AxisPosn", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "FaceCenter",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"EdgeCenter", "OnVertex", "OnEdge", or "OnFace">, "UParam:=" , <float>, "VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateFaceCS(*<Parameters>*,*<AttributesArray>*)
Python Example	oEditor.CreateFaceCS(["NAME:FaceCSParameters", ["NAME:Origin", "IsAttachedToEntity:=" , True, "EntityID:=" , 46, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=" , -1, "PositionType:=" , "FaceCenter", "UParam:=" , 0, "VParam:=" , 0, "XPosition:=" , "0", "YPosition:=" , "0", "ZPosition:=" , "0"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ],
  "MoveToEnd:="           , False,
  "FaceID:="              , 46,
  ["NAME:AxisPosn",
    "IsAttachedToEntity:="  , True,
    "EntityID:="            , 46,
    "FacetedBodyTriangleIndex:=", -1,
    "TriangleVertexIndex:=" , -1,
    "PositionType:="        , "OnFace",
    "UParam:="              , 0.487129134674319,
    "VParam:="              , 0.308528523557527,
    "XPosition:="           , "1292.27748080459mm",
    "YPosition:="           , "-814.882885865484mm",
    "ZPosition:="           , "0mm"
  ],
  "WhichAxis:="           , "X",
  "ZRotationAngle:="      , "0deg",
  "XOffset:="             , "0mm",
  "YOffset:="             , "0mm",
  "AutoAxis:="            , False
],
["NAME:Attributes",
  "Name:="                , "FaceCS1",
  "PartName:="            , "Rectangle1"
])
```

CreateGroup

Creates a group from objects specified in the history tree.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Modeler > Group > Create.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:GroupParameter",
"ParentGroupID:=" , <string>,
"Parts:=" , <string>,
"SubmodelInstances:=" , <string>,
"Groups:=" , <string>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreateGroup(<*Parameters*>)
Python Example	oEditor.CreateGroup(["NAME:GroupParameter", "ParentGroupID:=" , "Model", "Parts:=" , "Box1,Box2,Box3", "SubmodelInstances:=" , "", "Groups:=" , ""])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateNamedSelection

Creates a list of entities containing objects or faces (not both).

<table>
<tr><td>UI Access</td><td>Modeler > Named Selection > Create > [Object / Face] Selection</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.

```
["NAME:NamedSelectionParameters",
  "Type:=" , <string "Object" or "Face">,
  "Selection:=" , <string of object names or IDs>]
```


See GetObjectIDByName for returning object IDs.</td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>Structured array. See AttributesArray. CreateNamedSelection takes only the "Name" parameter and UDM ID.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreateNamedSelection(<*Parameters*>,<*AttributesArray*>)
Python Example	

```
namedSelectionName = oEditor.CreateNamedSelection(
  [
    "NAME:NamedSelectionParameters",
    "Type:=" , "Face",
    "Selection:=" , [7,9]
  ],
  [
    "NAME:Attributes",
    "Name:=" , "FaceSelection1",
    "UDM ID:=" , -1
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateObjectCS

Creates an object coordinate system from a selected object.

<table>
<tr><td>UI Access</td><td colspan="3">Modeler > Coordinate System > Create > Object > [Offset / Rotated / Both].</td></tr>
<tr><td rowspan="3">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:ObjectCSParameters",
  <OriginArray>
  "MoveToEnd:=" , <boolean>,
  "ReverseXAxis:=" , <boolean>,
  "ReverseYAxis:=" , <boolean>,
  <xAxisArray / xAxisPosArray>
  <yAxisArray / yAxisPosArray>]</pre>Note: xAxisArray and xAxisPosArray differ. Use xAxisArray for absolute position and xAxisPosArray for relative position. Do the same for yAxisArray and yAxisPosArray.</td></tr>
<tr><td><OriginArray></td><td>Array</td><td>Structured array.
<pre>["NAME:Origin",
  "IsAttachedToEntity:=" , <boolean>,
  "EntityID:=" , <integer>,
  "FacetedBodyTriangleIndex:=", <integer>,
  "TriangleVertexIndex:=" , <integer>,
  "PositionType:=" , <string "OnVertex",
  "EdgeCenter", "FaceCenter", "OnEdge", or "Abso-
  lutePosition">,</pre></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"UParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "VParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>] IsAttachedToEntity specifies whether the point is anchored. If True, provide UParam and VParam. If False, provide XPosition, YPosition, and ZPosition to provide fixed position. Pass "0" for unused parameters.
	<xAxisArray>	Array	Structured array for absolute position: ["NAME:xAxis", "DirectionType:=" , "AbsoluteDirection", "EdgeID:=" , <integer>, "FaceID:=" , <integer>, "xDirection:=" , <string>, "yDirection:=" , <string>, "zDirection:=" , <string>, "UParam:=" , <float>, "VParam:=" , <float>]
	<xAxisPosArray>	Array	Structured array for relative position: ["NAME:xAxisPos", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		"PositionType:=" , <string "OnVertex", "EdgeCenter", "FaceCenter", or "OnEdge">, "UParam:=" , <float>, "VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
<yAxisArray>	Array	Structured array for absolute position: ["NAME:yAxis", "DirectionType:=" , "AbsoluteDirection", "EdgeID:=" , <integer>, "FaceID:=" , <integer>, "xDirection:=" , <string>, "yDirection:=" , <string>, "zDirection:=" , <string>, "UParam:=" , <float>, "VParam:=" , <float>]
<yAxisPosArray>	Array	Structured array for relative position: ["NAME:yAxisPos", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "OnVertex", "EdgeCenter", "FaceCenter", or "OnEdge">, "UParam:=" , <float>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateObjectCS(<*Parameters*>,<*AttributesArray*>)
Python Example	oEditor.CreateObjectCS(["NAME:ObjectCSParameters", ["NAME:Origin", "IsAttachedToEntity:=" , True, "EntityID:=" , 59, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=" , -1, "PositionType:=" , "OnVertex", "UParam:=" , 0, "VParam:=" , 0, "XPosition:=" , "0", "YPosition:=" , "0", "ZPosition:=" , "0"], "MoveToEnd:=" , False, "ReverseXAxis:=" , False, "ReverseYAxis:=" , False, ["NAME:xAxis", "DirectionType:=" , "AbsoluteDirection",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "EdgeID:="              , -1,
    "FaceID:="              , -1,
    "xDirection:="          , "1",
    "yDirection:="          , "0",
    "zDirection:="          , "0",
    "UParam:="              , 0,
    "VParam:="              , 0
  ],
  ["NAME:yAxis",
    "DirectionType:="       , "AbsoluteDirection",
    "EdgeID:="              , -1,
    "FaceID:="              , -1,
    "xDirection:="          , "0",
    "yDirection:="          , "1",
    "zDirection:="          , "0",
    "UParam:="              , 0,
    "VParam:="              , 0
  ]
],
["NAME:Attributes",
  "Name:="                , "ObjectCS1",
  "PartName:="            , "Box2"
])
```

CreateObjectFromEdges

Creates an object from the specified object edge.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Modeler** > **Edge** > **Create Object From Edge**

Parameters

Name	Type	Description
<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

<*ParametersArray*>	Array	Structured array.
		`["NAME:Parameters",`
		`["NAME:BodyFromEdgeToParameters",`
		`"Edges:=" , <array containing integer edges>`
		`]`
		`]`

<*CreateGroupsForNewObjects*>	Array	Structured array. `["CreateGroupsForNewObjects:=", <boolean True to create groups for new objects; else False>]`

Return Value	None.

Python Syntax	CreateObjectFromEdges(<*SelectionsArray*>, <*ParametersArray*>, <*CreateGroupsForNewObjects*>)

Python Example

```
oEditor.CreateObjectFromEdges(
  ["NAME:Selections",
    "Selections:=", "Box2",
    "NewPartsModelFlag:=", "Model"
  ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``` ["NAME:Parameters", ["NAME:BodyFromEdgeToParameters", "Edges:=", [41]]], ["CreateGroupsForNewObjects:=", False]) ```

CreateObjectFromFace

Creates 2D objects from specified face(s).

UI Access	N/A
Parameters	See table below
Return Value	None.

Name	Type	Description
<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

<*Parameters*>	Array	Structured array.
		`["NAME:Parameters",`
		`<BodyFromFaceToParameters>]`

<*CreateGroupsForNewObjects*>	Array	Optional. Structured array. `["CreateGroupsForNewObjects:=", <boolean>]`

Python Syntax	CreateObjectFromFace (<*SelectionsArray*>,<*Parameters*>, <*CreateGroupsForNewObjects*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oEditor.CreateObjectFromFace(["NAME:Selections", "Selections:=" , "Box3", "NewPartsModelFlag:=" , "Model"], ["NAME:Parameters", ["NAME:BodyFromFaceToParameters", "FacesToDetach:=" , [68]]], ["CreateGroupsForNewObjects:=", False])

CreateObjectFromFaces

Creates 2D objects from specified face(s).

UI Access	**Modeler > Surface > Create Object from Face**		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.
	<Parameters>	Array	Structured array. ["NAME:Parameters", <BodyFromFaceToParameters>]
	<CreateGroupsForNewObjects>	Array	Structured array.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["CreateGroupsForNewObjects:=", <boolean>]
Return Value	None.		

Python Syntax	CreateObjectFromFaces (<*SelectionsArray*>,<*Parameters*>, <*CreateGroupsForNewObjects*>)

Python Example	`oEditor.CreateObjectFromFaces(`
	`  ["NAME:Selections",`
	`  "Selections:=" , "Box3",`
	`  "NewPartsModelFlag:=" , "Model"`
	`  ],`
	`  ["NAME:Parameters",`
	`  ["NAME:BodyFromFaceToParameters",`
	`  "FacesToDetach:=" , [68]`
	`  ]`
	`  ],`
	`  ["CreateGroupsForNewObjects:=", False`
	`  ])`

CreateRelativeCS

Creates a Relative Coordinate System.

UI Access	**Modeler** > **Coordinate System** > **Create** > **Relative CS** > [**Offset** / **Rotated** / **Both**].
Parameters	Name: <*Parameters*> — Type: Array — Description: Structured array. `["NAME:RelativeCSParameters",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Mode:=" , "Axis/Position", "OriginX:=" , <string>, "OriginY:=" , <string>, "OriginZ:=" , <string>, "XAxisXvec:=" , <string>, "XAxisYvec:=" , <string>, "XAxisZvec:=" , <string>, "YAxisXvec:=" , <string>, "YAxisYvec:=" , <string>, "YAxisZvec:=" , <string>]
	<AttributesArray>	Array	Structured array. See: AttributesArray. CreateRelativeCS supports only the "Name" parameter.
Return Value	None.		

Python Syntax	CreateRelativeCS(*<Parameters>*,*<AttributesArray>*)
Python Example	oEditor.CreateRelativeCS(["NAME:RelativeCSParameters", "Mode:=" , "Axis/Position", "OriginX:=" , "0.62mm", "OriginY:=" , "-0.7mm", "OriginZ:=" , "0mm", "XAxisXvec:=" , "1mm", "XAxisYvec:=" , "0mm", "XAxisZvec:=" , "0mm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "YAxisXvec:="           , "0mm",
    "YAxisYvec:="           , "1mm",
    "YAxisZvec:="           , "0mm"
  ],
  ["NAME:Attributes",
    "Name:="                , "RelativeCS1"
  ])
```

DeleteEmptyGroups

Deletes group(s) from the history tree.

UI Access	**Modeler** > **Group** > **Delete Empty**.		
Parameters	Name	Type	Description
	<*Parameters*>	Array	Structured array. `["Groups:=" , <array of string group IDs>]`
Return Value	None.		

Python Syntax	DeleteEmptyGroups(<*Parameters*>)

Python Example	`oEditor.DeleteEmptyGroups([`
	`  "Groups:=", ["Group1","Group2","Group3"]`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteLastOperation

Deletes the last operation performed on the specified object(s).

UI Access	**Modeler** > **Delete Last Operation**.
Parameters	Name / Type / Description <*SelectionsArray*> / Array / Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	DeleteLastOperation(<*SelectionsArray*>)

Python Example	`oEditor.DeleteLastOperation([`
	`  "NAME:Selections",`
	`  "Selections:=", "Box3",`
	`  "NewPartsModelFlag:=", "Model"`
	`])`

DeleteOperation

Deletes specified operation performed on a selected object.

UI Access	Select an operation in the project tree, then press **Delete** on the keyboard.

Parameters	Name / Type / Description
	<*ParamsArray*> / Array / Structured array.
	`["NAME:Parameters",`
	`  ["NAME:PartOperations",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:<PartName>", "OperationIndices:=", <array of operation indices>]], ["NAME:UDMOperations"]]
Return Value	None.		

Python Syntax	DeleteOperation(<*ParamsArray*>)
Python Example	oEditor.DeleteOperation(["NAME:Parameters", ["NAME:PartOperations", ["NAME:Coil_0", "OperationIndices:=", [1]]], ["NAME:UDMOperations"]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DetachEdges

Detaches the specified edge(s) from an object.

<table>
<tr><td>UI Access</td><td>Modeler > Edge > Detach Edges.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<code>["NAME:Parameters",</code>
<code> <DetachEdgesArray>]</code></td></tr>
<tr><td><DetachEdgesArray></td><td>Array</td><td>Structured array.
<code>["NAME:DetachEdgesToParameters",</code>
<code> "EdgesToDetach:=" , <array containing integer edge IDs>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	DetachEdges(<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.DetachEdges(`
	`["NAME:Selections",`
	`"Selections:=", "Rectangle1",`
	`"NewPartsModelFlag:=", "Model"`
	`],`
	`["NAME:Parameters",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:DetachEdgesToParameters", "EdgesToDetach:=", [18,17]]])

DetachFaces

Detaches the specified face(s) from an object.

UI Access	**Modeler** > **Surface** > **Detach Faces**.
Parameters	Name / Type / Description *<SelectionsArray>* / Array / Structured array. See: SelectionsArray. *<Parameters>* / Array / Structured array. ["NAME:Parameters", <DetachFacesArray>] *<DetachFacesArray>* / Array / Structured array. ["NAME:DetachFacesToParameters", "FacesToDetach:=" , <array containing integer face IDs>]
Return Value	None.

Python Syntax	DetachFaces(*<SelectionsArray>*, *<Parameters>*)
Python Example	oEditor.DetachFaces(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``
	["NAME:Selections",
	  "Selections:=", "Box3",
	  "NewPartsModelFlag:=", "Model"
	],
	["NAME:Parameters",
	  ["NAME:DetachFacesToParameters",
	  "FacesToDetach:=", [68,67]
	  ]
	])
	``

EditEntityList

Modifies an entity list.

> **Warning:** As of the Ansys Electronics Desktop25r2 release, this command has been deprecated. It will be removed in a future release. Please use the EditNamedSelection command instead.

UI Access	**Modeler** > **Named Selection** > **Reassign**.		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.

	<Parameters>	Array	Structured array.
			`["NAME:GeometryEntityListParameters",`
			`  "EntityType:=" , <string "Object" or "Face">,`
			`  "EntityList:=" , <string list>`
			`]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	EditEntityList(<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.EditEntityList(`
	`  ["NAME:Selections",`
	`  "Selections:="  , "Objectlist1"`
	`  ],`
	`  ["NAME:GeometryEntityListParameters",`
	`  "EntityType:="  , "Object",`
	`  "EntityList:="  , "Box1, Box2, Box3"`
	`  ])`

EditFaceCS

Recreates an existing face coordinate system. See: CreateFaceCS.

UI Access	**Modeler** > **Coordinate System** > **Edit**.

Parameters	Name: <*Parameters*> — Type: Array — Description: Structured array.
	`["NAME:FaceCSParameters",`
	`  <OriginArray>,`
	`  "MoveToEnd:=", <boolean>,`
	`  "FaceID:=", <integer>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		<AxisPosnArray>, "WhichAxis:=" , <string "X", "Y", or "Z">, "ZRotationAngle:=" , <string>, "XOffset:=" , <string>, "YOffset:=" , <string>, "AutoAxis:=" , <boolean>]
<OriginArray>	Array	Structured array. ["NAME:Origin", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "FaceCenter", "EdgeCenter", "OnVertex", "OnEdge", or "OnFace">, "UParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "VParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>] IsAttachedToEntity specifies whether the point is anchored to a vertex, edge, or face. If True, provide UParam and VParam. If False, provide XPosition, YPosition, and ZPosition to provide fixed position. Pass "0" for unused parameters.
<AxisPosnArray>	Array	Structured array. ["NAME:AxisPosn",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "FaceCenter", "EdgeCenter", "OnVertex", "OnEdge", or "OnFace">, "UParam:=" , <float>, "VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray. Use to select the coordinate system to edit.
Return Value	None.		

Python Syntax	EditFaceCS (<*Parameters*>, <*AttributesArray*>)
Python Example	oEditor.EditFaceCS(["NAME:FaceCSParameters", ["NAME:Origin", "IsAttachedToEntity:=" , True, "EntityID:=" , 12, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=" , -1, "PositionType:=" , "FaceCenter", "UParam:=" , 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "VParam:="              , 0,
    "XPosition:="           , "0",
    "YPosition:="           , "0",
    "ZPosition:="           , "0"
  ],
  "MoveToEnd:="           , False,
  "FaceID:="              , 12,
  ["NAME:AxisPosn",
    "IsAttachedToEntity:="  , True,
    "EntityID:="            , 12,
    "FacetedBodyTriangleIndex:=", -1,
    "TriangleVertexIndex:=" , -1,
    "PositionType:=" , "OnFace",
    "UParam:="      , 0.62951717774066,
    "VParam:="      , 0.226514925559344,
    "XPosition:="   , "1200mm",
    "YPosition:="   , "-354.697014888131mm",
    "ZPosition:="   , "125.903435548132mm"
  ],
  "WhichAxis:="           , "X",
  "ZRotationAngle:="      , "0deg",
  "XOffset:="             , "0mm",
  "YOffset:="             , "0mm",
  "AutoAxis:="            , False
],
["NAME:Attributes",
"Name:="                , "FaceCS1",
"PartName:="            , "Box1"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

EditNamedSelection

Modifies an entity list.

UI Access	**Modeler > Named Selection > Reassign**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:NamedSelectionParameters", "Type:=" , <string "Object" or "Face">, "Selection:=" , <string of object names or IDs>) "Mode:=" , <defines action to named selection, includes "Add", "Remove", and "Reassign">]</td></tr></table>
Return Value	None.

Python Syntax	EditNamedSelection(<*SelectionsArray*>, <*Parameters*>)
Python Example	oEditor.EditNamedSelection(["NAME:Selections", "Selections:=" , "FaceSelection1"], [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NAME:NamedSelectionParameters", "Type:=" , "Face", "Selection:=" , [12] "Mode:", "Remove"])

EditObjectCS

Edits an existing object coordinate system. See: CreateObjectCS.

UI Access	**Modeler** > **Coordinate System** > **Edit**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:ObjectCSParameters", <OriginArray> "MoveToEnd:=" , <boolean>, "ReverseXAxis:=" , <boolean>, "ReverseYAxis:=" , <boolean>, <xAxisArray / xAxisPosArray> <yAxisArray / yAxisPosArray>] **Note:** xAxisArray and xAxisPosArray differ. Use xAxisArray for absolute position and xAxisPosArray for relative position. Do the same for yAxisArray and yAxisPosArray.
	<OriginArray>	Array	Structured array. ["NAME:Origin", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"PositionType:=" , <string "OnVertex", "EdgeCenter", "FaceCenter", "OnEdge", or "AbsolutePosition">, "UParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "VParam:=" , <float between 0 and 1 representing the relative position of the point on the edge or face>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>] IsAttachedToEntity specifies whether the point is anchored. If True, provide UParam and VParam. If False, provide XPosition, YPosition, and ZPosition to provide fixed position. Pass "0" for unused parameters.
	<*xAxisArray*>	Array	Structured array for absolute position: ["NAME:xAxis", "DirectionType:=" , "AbsoluteDirection", "EdgeID:=" , <integer>, "FaceID:=" , <integer>, "xDirection:=" , <string>, "yDirection:=" , <string>, "zDirection:=" , <string>, "UParam:=" , <float>, "VParam:=" , <float>]
	<*xAxisPosArray*>	Array	Structured array for relative position: ["NAME:xAxisPos", "IsAttachedToEntity:=" , <boolean>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "OnVertex", "EdgeCenter", "FaceCenter", or "OnEdge">, "UParam:=" , <float>, "VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
	<yAxisArray>	Array	Structured array for absolute position: ["NAME:yAxis", "DirectionType:=" , "AbsoluteDirection", "EdgeID:=" , <integer>, "FaceID:=" , <integer>, "xDirection:=" , <string>, "yDirection:=" , <string>, "zDirection:=" , <string>, "UParam:=" , <float>, "VParam:=" , <float>]
	<yAxisPosArray>	Array	Structured array for relative position: ["NAME:yAxisPos", "IsAttachedToEntity:=" , <boolean>, "EntityID:=" , <integer>, "FacetedBodyTriangleIndex:=", <integer>, "TriangleVertexIndex:=" , <integer>, "PositionType:=" , <string "OnVertex", "EdgeCenter", "FaceCenter", or "OnEdge">, "UParam:=" , <float>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"VParam:=" , <float>, "XPosition:=" , <string>, "YPosition:=" , <string>, "ZPosition:=" , <string>]
	<*AttributesArray*>	Array	Structured array. See: AttributesArray. Use to select the coordinate system to edit.
Return Value	None.		

Python Syntax	EditObjectCS(<*Parameters*>,<*AttributesArray*>)
Python Example	oEditor.EditObjectCS(["NAME:ObjectCSParameters", ["NAME:Origin", "IsAttachedToEntity:=" , True, "EntityID:=" , 59, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=" , -1, "PositionType:=" , "OnVertex", "UParam:=" , 0, "VParam:=" , 0, "XPosition:=" , "0", "YPosition:=" , "0", "ZPosition:=" , "0"], "MoveToEnd:=" , False, "ReverseXAxis:=" , False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "ReverseYAxis:="        , False,
  ["NAME:xAxis",
    "DirectionType:="     , "AbsoluteDirection",
    "EdgeID:="            , -1,
    "FaceID:="            , -1,
    "xDirection:="        , "1",
    "yDirection:="        , "0",
    "zDirection:="        , "0",
    "UParam:="            , 0,
    "VParam:="            , 0
  ],
  ["NAME:yAxis",
    "DirectionType:="     , "AbsoluteDirection",
    "EdgeID:="            , -1,
    "FaceID:="            , -1,
    "xDirection:="        , "0",
    "yDirection:="        , "1",
    "zDirection:="        , "0",
    "UParam:="            , 0,
    "VParam:="            , 0
  ]
],
["NAME:Attributes",
  "Name:="              , "ObjectCS1",
  "PartName:="          , "Box2"
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditRelativeCS

Edits an existing Relative Coordinate System. See: CreateRelativeCS.

<table>
<tr><td>UI Access</td><td colspan="3">Modeler > Coordinate System > Edit.</td></tr>
<tr><td rowspan="3">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:RelativeCSParameters",
  "Mode:=" , "Axis/Position",
  "OriginX:=" , <string>,
  "OriginY:=" , <string>,
  "OriginZ:=" , <string>,
  "XAxisXvec:=" , <string>,
  "XAxisYvec:=" , <string>,
  "XAxisZvec:=" , <string>,
  "YAxisXvec:=" , <string>,
  "YAxisYvec:=" , <string>,
  "YAxisZvec:=" , <string>]</pre></td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>Structured array. See: AttributesArray. Use to select the coordinate system to edit.</td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditRelativeCS(<*Parameters*>,<*AttributesArray*>)
Python Example	oEditor.EditRelativeCS(["NAME:RelativeCSParameters", "Mode:=" , "Axis/Position", "OriginX:=" , "0.62mm", "OriginY:=" , "-0.7mm", "OriginZ:=" , "0mm", "XAxisXvec:=" , "1mm", "XAxisYvec:=" , "0mm", "XAxisZvec:=" , "0mm", "YAxisXvec:=" , "0mm", "YAxisYvec:=" , "1mm", "YAxisZvec:=" , "0mm"], ["NAME:Attributes", "Name:=" , "RelativeCS1"])

Export

Exports the model to a file.

> **Note:** This script does not export image file types or GDSII files. See: ExportModelImageToFile and ExportGDSII.

UI Access	**Modeler** > **Export**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><*Parameters*></td><td>Array</td><td>Structured array.
["NAME:ExportParameters",
"AllowRegionDependentPartSelectionForPMLCreation:=", <boolean>,
"AllowRegionSelectionForPMLCreation:=", <boolean>,
"Selections:=" , <string list>,
"File Name:=" , <string filepath>,
"Major Version:=" , <integer (-1 if not applicable)>,
"Minor Version:=" , <integer (-1 if not applicable)>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	Export(<*Parameters*>)
Python Example	oEditor.Export(["NAME:ExportParameters", "AllowRegionDependentPartSelectionForPMLCreation:=", True, "AllowRegionSelectionForPMLCreation:=", True, "Selections:=" , "Box1,Box2,Box3", "File Name:=" , "C:/Users/jdoe/Desktop/export.sab",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Major Version:="       , 25,
  "Minor Version:="       , 0
])
```

ExportModelImageToFile

Exports the model as an image file (*.avz, *.bmp, *.gif, *.jpeg, *.tiff, *.wrl). In Release 23.1, this command is fully supports -ng (non-graphical) mode. To export to Ensight use *.avz. For export to Ensight in -ng mode, the corresponding version of Ensight must be installed. On Linux, it might need manual set environment variable AWP_ROOT212 to its installation path, e.g. AWP_ROOT212-2=/installations/ansys_inc/v212/ for AnsysEDT v21.2 and Ensight 21.2.

ExportModelImageToFile exports a model image with background type and color that respect the AEDT color scheme by default. You can specify the background type and color with following parameters:

Parameter Name	Description	Parameter Value
BackgroundType	Choose one out of four types of background	Default Plain LinearGradient RadialGradient
BackgroundColor	Plain: Background color Lin-earGradient/RadialGradient: Background start color	3 integers with a range of [0,255] that represent red, green, and blue respectively
BackgroundContrastColor	Plain: Ignored LinearGradient/RadialGradient: Back-ground end color	3 integers with a range of [0,255] that represent red, green, and blue respectively

Note: Current scripts will not be affected, the image will be exported with default background color as it always does

If no BackgroundType is specified, or BackgroundType is Default, BackgroundColor and BackgroundContrastColor will be ignored, and the image will be exported with default background color

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportModelImageToFile supports export overlay of polar plot 3D with existing transformation (scaling, rotation and translation) in -ng (non-graphical) mode.

<table>
<tr><td>UI Access</td><td>Modeler > Export.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><path></td><td>String</td><td>Full file path including extension.</td></tr>
<tr><td><width></td><td>Integer</td><td>Width in pixels (use 0 for default).</td></tr>
<tr><td><height></td><td>Integer</td><td>Height in pixels (use 0 for default).</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:SaveImageParams",
  "ShowAxis:=" , <string containing boolean>,
  "ShowGrid:=" , <string containing boolean>,
  "ShowRuler:=" , <string containing boolean>,
  "ShowRegion:=" , <string>,
  "Selections:=" , <string>,
  "FieldPlotSelections:=", <string>' # Comma delimited string. #Use to set which field plot to show.
  "FitToSelections:=" , "",</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: "FitToSelections" specify geometry objects for the "Fit" operation.
			`"FitToFieldPlotSelection:=" , ""`
			Note: "FitToFieldPlotSelections" specifies field plots for the "Fit" operation.
			`"AutoFit:=" , "True",`
			Note: If FitToSlections or FitToFieldPlotSelections are used , then AutoFit is True, it makes sure color key does not overlap field plot. It is False by default. If neither is used, then "AutoFit" will "Fit" to full model.
			`"Orientation:=" , <string>`
			`"ShowOrientationGadget:=" , <False>]`

Return Value	None.		

Python Syntax	ExportModelImageToFile(<*path*> <*width*> <*height*> <*Parameters*>)

Python Example	`oEditor.ExportModelImageToFile`
	`("D:/Image.png", 1920, 1080,`
	`[`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "NAME:SaveImageParams",
    "ShowAxis:=" , "True",
    "ShowGrid:=" , "True",
    "ShowRuler:=" , "True",
    "ShowRegion:=" , "Default",
    "Selections:=" , "",
    "FieldPlotSelections:=" , "",
    "FitToSelections:=" , "",
    "FitToFieldPlotSelections:=", "",
    "AutoFit:=" , "True",
    "Orientation:=" , ""
])
```

ExportModelMeshToFile

Exports geometry model to a 3D model file (e.g. *.obj, *.wrl, etc.).

UI Access	N/A
Parameters	

Name	Type	Description
<filePath>	String	Full file path, including extension *.obj, *.wrl, etc
<selections>	Array	Selected parts.

Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportModelMeshToFile <*filePath*>, <*selections*>)
Python Example	`oEditor.ExportModelMeshToFile("E:/MyDir/scriptRun/2Selected-ng.obj", ["BotCover-","AveragingVolumeAtPeakRMSEfieldLocation"])`

Fillet

Performs a fillet on specified edge(s).

UI Access: **Modeler** > **Fillet**.

Parameters

Name	Type	Description
<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

<*Parameters*>	Array	Structured array.
		`["NAME:Parameters",`
		`<FilletParametersArray>]`

<*FilletParametersArray*>	Array	Structured array.
		`["NAME:FilletParameters",`
		`"Edges:=" , <array containing integer edge IDs>,`
		`"Vertices:=" , <empty array>,`
		`"Radius:=" , <string>,`
		`"Setback:=" , <string>]`

Return Value: None.

Python Syntax	Fillet(<*SelectionsArray*>, <*Parameters*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.Fillet(`
	`["NAME:Selections",`
	`  "Selections:="  , "Box1",`
	`  "NewPartsModelFlag:="  , "Model"`
	`],`
	`["NAME:Parameters",`
	`  ["NAME:FilletParameters",`
	`  "Edges:="  , [13],`
	`  "Vertices:="  , [],`
	`  "Radius:="  , "1mm",`
	`  "Setback:="  , "0mm"`
	`  ]`
	`])`

FlattenGroup

Flattens a specified history tree group.

UI Access	**Modeler** > **Group** > **Flatten**.
Parameters	Name: *<GroupID>*; Type: Array; Description: Structured array. `["Groups:=", [<string list of group IDs>]]`
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	FlattenGroup (<*GroupID*>)
Python Example	`oEditor.FlattenGroup(["Groups:=", ["Group1"]])`

GenerateHistory

Generates the history for specified 1D object(s).

UI Access	**Modeler** > **Generate History**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr></table>
Return Value	None.

Python Syntax	GenerateHistory (<*SelectionsArray*>)

Python Example	`oEditor.GenerateHistory(`
	`["NAME:Selections",`
	`  "Selections:="  , "Polyline1",`
	`  "NewPartsModelFlag:="  , "Model",`
	`  "UseCurrentCS:="  , True`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HealObject

Heals an imported object.

<table>
<tr><td>UI Access</td><td colspan="3">Modeler > Model Preparation > Heal.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:ObjectHealingParameters",
  "Version:=" , <integer>,
  "AutoHeal:=" , <boolean>,
  "TolerantStitch:=" , <boolean>,
  "SimplifyGeom:=" , <boolean>,
  "TightenGaps:=" , <boolean>,
  "HealToSolid:=" , <boolean>,
  "StopAfterFirstStitchError:=", <boolean>,
  "MaxStitchTol:=" , <float>,
  "ExplodeAndStitch:=" , <boolean>,
  "GeomSimplificationTol:=", <integer>,
  "MaximumGeneratedRadiusForSimplification:=",
  <integer>,
  "SimplifyType:=" , <integer>,
  "TightenGapsWidth:=" , <float>,
  "RemoveSliverFaces:=" , <boolean>,
  "RemoveSmallEdges:=" , <boolean>,
  "RemoveSmallFaces:=" , <boolean>,
  "SliverFaceTol:=" , <integer>,</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"SmallEdgeTol:=" , <integer>, "SmallFaceAreaTol:=" , <integer>, "BoundingBoxScaleFactor:=", <integer>, "RemoveHoles:=" , <boolean>, "RemoveChamfers:=" , <boolean>, "RemoveBlends:=" , <boolean>, "HoleRadiusTol:=" , <integer>, "ChamferWidthTol:=" , <integer>, "BlendRadiusTol:=" , <integer>, "AllowableSurfaceAreaChange:=", <integer>, "AllowableVolumeChange:=", <integer>]
Return Value	None.		

Python Syntax	HealObject(<*SelectionsArray*>, <*Parameters*>)

Python Example	oEditor.HealObject(
	["NAME:Selections",
	  "Selections:=" , "Box1",
	  "NewPartsModelFlag:=" , "Model"
	],
	["NAME:ObjectHealingParameters",
	  "Version:=" , 1,
	  "AutoHeal:=" , True,
	  "TolerantStitch:=" , True,
	  "SimplifyGeom:=" , True,
	  "TightenGaps:=" , True,
	  "HealToSolid:=" , False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "StopAfterFirstStitchError:=", False,
  "MaxStitchTol:="        , 0.001,
  "ExplodeAndStitch:="    , True,
  "GeomSimplificationTol:=", -1,
  "MaximumGeneratedRadiusForSimplification:=", -1,
  "SimplifyType:="        , 2,
  "TightenGapsWidth:="    , 1E-06,
  "RemoveSliverFaces:="   , False,
  "RemoveSmallEdges:="    , False,
  "RemoveSmallFaces:="    , False,
  "SliverFaceTol:="       , 0,
  "SmallEdgeTol:="        , 0,
  "SmallFaceAreaTol:="    , 0,
  "BoundingBoxScaleFactor:=", 1250,
  "RemoveHoles:="         , False,
  "RemoveChamfers:="      , False,
  "RemoveBlends:="        , False,
  "HoleRadiusTol:="       , 0,
  "ChamferWidthTol:="     , 0,
  "BlendRadiusTol:="      , 0,
  "AllowableSurfaceAreaChange:=", 5,
  "AllowableVolumeChange:=", 5
])
```

Import

Imports a 3D model file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

This script does not import DXF or GDSII models. See: ImportDXF and ImportGDSII.

UI Access	**Modeler** > **Import**.

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:NativeBodyParameters",`
		`"HealOption:=" , <integer>,`
		`"Options:=" , <string>,`
		`"FileType:=" , <string "UnRecognized">,`
		`"MaxStitchTol:=" , <integer>,`
		`"ImportFreeSurfaces:=" , <boolean>,`
		`"GroupByAssembly:=" , <boolean>,`
		`"CreateGroup:=" , <boolean>,`
		`"STLFileUnit:=" , <string>,`
		`"MergeFacesAngle:=" , <float>,`
		`"HealSTL:=" , <boolean>,`
		`"ReduceSTL:=" , <boolean>,`
		`"ReduceMaxError:=" , <integer>,`
		`"ReducePercentage:=" , <integer>,`
		`"PointCoincidenceTol:=" , <float>,`
		`"CreateLightweightPart:=", <boolean>,`
		`"ImportMaterialNames:=" , <boolean>,`
		`"SeparateDisjointLumps:=", <boolean>,`
		`"SourceFile:=" , <string>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	Import(<*Parameters*>)

Python Example	`oEditor.Import(`
	`["NAME:NativeBodyParameters",`
	`  "HealOption:="  , 0,`
	`  "Options:="  , "-1",`
	`  "FileType:="  , "UnRecognized",`
	`  "MaxStitchTol:="  , -1,`
	`  "ImportFreeSurfaces:="  , False,`
	`  "GroupByAssembly:="  , False,`
	`  "CreateGroup:="  , True,`
	`  "STLFileUnit:="  , "Auto",`
	`  "MergeFacesAngle:="  , 0.02,`
	`  "HealSTL:="  , False,`
	`  "ReduceSTL:="  , False,`
	`  "ReduceMaxError:="  , 0,`
	`  "ReducePercentage:="  , 100,`
	`  "PointCoincidenceTol:=" , 1E-06,`
	`  "CreateLightweightPart:=", False,`
	`  "ImportMaterialNames:=" , False,`
	`  "SeparateDisjointLumps:=", False,`
	`  "SourceFile:="  , "C:\\Users\\jdoe\\Desktop\\export.model"`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportDXF [Modeler]

Imports a DXF model file.

<table>
<tr><td>UI Access</td><td>Modeler > Import.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:options",
"FileName:=" , <string>,
"Scale:=" , <float>,
"AutoDetectClosed:=" , <boolean>,
"SelfStitch:=" , <boolean>,
"DefeatureGeometry:=" , <boolean>,
"DefeatureDistance:=" , <integer>,
"RoundCoordinates:=" , <boolean>,
"RoundNumDigits:=" , <integer>,
"WritePolyWithWidthAsFilledPoly:=", <boolean>,
"ImportMethod:=" , <integer>,
"2DSheetBodies:=" , <boolean>,
<layersArray>]</td></tr>
<tr><td><layersArray></td><td>Array</td><td>Structured array.
["NAME:LayerInfo",
<layer>, <layer>, <layer>,...]</td></tr>
<tr><td><layer></td><td>Array</td><td>Structured array.
["NAME:<layerName>",
"source:=" , <string>,
"display_source:=" , <string>,
"import:=" , <boolean>,</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"dest:=" , <string>, "dest_selected:=" , <boolean>, "layer_type:=" , <string>]
Return Value	None.		

Python Syntax	ImportDXF(<*Parameters*>)
Python Example	oEditor.ImportDXF(["NAME:options", "FileName:=", "C:/Users/jdoe/Desktop/export.dxf", "Scale:=" , 0.001, "AutoDetectClosed:=" , True, "SelfStitch:=" , True, "DefeatureGeometry:=" , False, "DefeatureDistance:=" , 0, "RoundCoordinates:=" , False, "RoundNumDigits:=" , 4, "WritePolyWithWidthAsFilledPoly:=", False, "ImportMethod:=" , 1, "2DSheetBodies:=" , False, ["NAME:LayerInfo", ["NAME:0", "source:=" , "0", "display_source:=" , "0", "import:=" , True, "dest:=" , "0",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ],
    ["NAME:LAYER_1",
    "source:="              , "LAYER_1",
    "display_source:="      , "LAYER_1",
    "import:="              , True,
    "dest:="                , "LAYER_1",
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ],
    ["NAME:LAYER_2",
    "source:="              , "LAYER_2",
    "display_source:="      , "LAYER_2",
    "import:="              , True,
    "dest:="                , "LAYER_2",
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ]
  ]
])
```

ImportFromClipboard

Imports a model from clipboard.

UI Access	**Modeler** > **Import From Clipboard**.
Parameters	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ImportFromClipboard ()
Python Example	`oEditor.ImportFromClipboard()`

ImportGDSII [Modeler]

Imports a GDSII model file.

UI Access: **Modeler** > **Import**.

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:options",`
		`  "FileName:=" , <string>,`
		`  "FlattenHierarchy:=" , <boolean>,`
		`  "ImportMethod:=" , <integer>,`
		`  <layerMapArray>, <layerMapArray>, <layerMapArray>,...`
		`"OrderMap:=" ,`
		`  [ "entry:=" ,`
		`  <entry>, <entry>, <entry>,...`
		`  ]`
		`])`

<layerMapArray>	Array	Structured array.
		`["NAME:LayerMapInfo",`
		`  "LayerNum:=" , <integer>,`
		`  "DestLayer:=" , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"layer_type:=" , <string>]
	<*entry*>	Array	Structured array. ["order:=" , <integer LayerNum>, "layer:=" , <string DestLayer>]
Return Value	None.		

Python Syntax	ImportGDSII(<*Parameters*>)
Python Example	oEditor.ImportGDSII(["NAME:options", "FileName:=", "C:/Users/coil2.gds", "FlattenHierarchy:=" , True, "ImportMethod:=" , 1, ["NAME:LayerMap", ["NAME:LayerMapInfo", "LayerNum:=" , 12, "DestLayer:=" , "Signal12", "layer_type:=" , "signal"], ["NAME:LayerMapInfo", "LayerNum:=" , 13, "DestLayer:=" , "Signal13", "layer_type:=" , "signal"], ["NAME:LayerMapInfo", "LayerNum:=" , 14, "DestLayer:=" , "Signal14",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
     "layer_type:="          , "signal"
     ]
   ],
"OrderMap:=",
   [
   "entry:=",
     ["order:=", 0,  "layer:=", "Signal12"],
   "entry:=",
     ["order:=", 1,  "layer:=", "Signal13"],
   "entry:=",
     ["order:=", 2,  "layer:=", "Signal14"]
   ]
])
```

Imprint

Imprints the geometry of one object upon another.

UI Access	**Modeler** > **Boolean** > **Imprint...**.		
Parameters	Name	Type	Description

	<*Selections*>	Array	Structured array.
			`["NAME:Selections",`
			`"Blank Parts:=", <string, object name>,`
			`"Tool Parts:=", <string, object name>]`

	<*Parameters*>	Array	Structured array.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:ImprintParameters", "KeepOriginals:=", <boolean>]
Return Value	None.		

Python Syntax	Imprint (<*Selections*>, <*Parameters*>)
Python Example	oEditor.Imprint(["NAME:Selections", "Blank Parts:=", "Cylinder1", "Tool Parts:=", "Box1"], ["NAME:ImprintParameters", "KeepOriginals:=", False])

ImprintProjection

Projects the form of a sheet object onto the face or faces of another object (either solid or sheet).

UI Access	**Modeler** > **Boolean** > **Imprint Projection**.		
Parameters	Name	Type	Description
	<*Selections*>	Array	Structured array. ["NAME:Selections", "Selections:=", <string, selected objects>]
	<*Parameters*>	Array	Structured array.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:ImprintProjectionParameters", "KeepOriginals:=", <boolean>, "NormalProjection:=", <boolean>, "Distance:=" , <float>, "DirectionX:=" , <float>, "DirectionY:=" , <float>, "DirectionZ:=" , <float>]
Return Value	None.		

Python Syntax	ImprintProjection (<*Selections*>, <*Parameters*>)
Python Example	oEditor.ImprintProjection(["NAME:Selections", "Selections:=", "Rectangle1,Cylinder1"], ["NAME:ImprintProjectionParameters", "KeepOriginals:=", False, "NormalProjection:=", True, "Distance:=", "1.36014705087354mm", "DirectionX:=", "0.882257546512569", "DirectionY:=", "0.294085848837523",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"DirectionZ:=", "0.367607311046904"])

Intersect

Intersects specified objects.

UI Access	**Modeler** > **Boolean** > **Intersect**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. List of objects to Boolean; see SelectionsArray.</td></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:IntersectParameters", "KeepOriginals:=" , <boolean True to keep original objects; False to delete>]</td></tr></table>
Return Value	None.

Python Syntax	Intersect(<*SelectionsArray*>, <*Parameters*>)
Python Example	oEditor.Intersect(["NAME:Selections", "Selections:=", "Rectangle1,Rectangle2"], ["NAME:IntersectParameters", "KeepOriginals:=", False])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

MoveCStoEnd

Moves a specified Object Coordinate System to the end of the History tree.

UI Access	**Modeler** > **Coordinate System** > **Move CS to End**.
Parameters	Name: *<SelectionsArray>* — Type: Array — Description: Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	MoveCStoEnd (<*SelectionsArray*>)

Python Example	`oEditor.MoveCSToEnd(`
	`["NAME:Selections",`
	`  "Selections:="  , "ObjectCS1"`
	`])`

MoveEntityToGroup

Moves a specified entity or entities to a specified group.

UI Access	Drag item into the group in the history tree.
Parameters	Name: *<Objects>* — Type: Array — Description: Structured array. Selects the entity/entities to move.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["Objects:=" , <array containing string object IDs>]
	<*MoveEntityToGroup*>	Array	Structured array. ["ParentGroup:=" , <string group name>]
Return Value	None.		

Python Syntax	MoveEntityToGroup (<*Objects*>, <*MoveEntityToGroup*>)
Python Example	oEditor.MoveEntityToGroup(["Objects:=", ["Box3"]], ["ParentGroup:=", "Box_Group"])

MoveFaces

Moves the specified faces along normal or along a vector.

UI Access	**Modeler** > **Surface** > **Move Faces** > **Along** [**Normal**/**Vector**].		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
	<*MoveFacesParametersArray*>	Array	Structured array. ["NAME:Parameters", <FacesOfOneObjToMove>, <FacesOfOneObjToMove>, ...]
	<*FacesOfOneObjToMove*>	Array	Structured array. ["Name:MoveFacesParameters",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"MoveAlongNormalFlag:=", <boolean>, "OffsetDistance:=", <string>, "MoveVectorX:=", <string>, "MoveVectorY:=", <string>, "MoveVectorZ:=", <string>, "FacesToMove:=", <array>] MoveAlongNormalFlag specifies whether to move along the face normal or along a vector. If false, provide the MoveVectorX, MoveVectorY, and MoveVectorZ parameters. FacesToMove is an array of integers (the IDs of the faces to move).
Return Value	None		

Python Syntax	MoveFaces (<*SelectionsArray*>, <*MoveFacesParametersArray*>)
Python Example	oEditor.MoveFaces(["NAME:Selections", "Selections:=" , "Rectangle1", "NewPartsModelFlag:=" , "Model"], ["NAME:Parameters", ["NAME:MoveFacesParameters",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "MoveAlongNormalFlag:="    , True,
            "OffsetDistance:="    , "1mm",
            "MoveVectorX:="        , "0mm",
            "MoveVectorY:="        , "0mm",
            "MoveVectorZ:="        , "0mm",
            "FacesToMove:="        , [183]
        ]
    ])
```

ProjectSheet

Project a sheet object, typically for modeling thin conformal deposits. Typically followed by **Thicken Sheet**.

UI Access	Click **Modeler** > **Surface** > **Project Sheet**.
Parameters	Name / Type / Description <*SelectionsArray*> / Array / Structured array. See: SelectionsArray. <*Parameters*> / Array / Array containing string. `["NAME:ProjectSheetParameters"]`
Return Value	None.

Python Syntax	ProjectSheet (<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.ProjectSheet(`
	`['NAME:Selections',`
	`'Selections:=','Box1,Box2,Polyline1'],`
	`['NAME:ProjectSheetParameters']`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	)

PurgeHistory

Purges the history of a specified object.

UI Access	**Modeler** > **Purge History**.
Parameters	Name | Type | Description <*SelectionsArray*> | Array | Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	PurgeHistory (<*SelectionsArray*>)

Python Example	`oEditor.PurgeHistory(`
	`["NAME:Selections",`
	`  "Selections:=", "Box2",`
	`  "NewPartsModelFlag:=", "Model"`
	`])`

ReplaceWith3DComponent

Replaces the selection with a 3D component.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	None.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*CreateData*></td><td>Array</td><td>Structured array.</td></tr><tr><td><*DesignData*></td><td>Array</td><td>Structured array.</td></tr><tr><td><*ImageFile*></td><td>Array</td><td>Structured array.</td></tr></table>
Return Value	None.

Python Syntax	ReplaceWith3DComponent(<*CreateData*>, <*DesignData*>, <*ImageFile*>)

Python Example	<pre>oEditor.ReplaceWith3DComponent(
	  ["NAME:CreateData",
	  "ComponentName:=", "CoaxBend",
	  "Company:=", "",
	  "Company URL:=", "",
	  "Model Number:=", "",
	  "Help URL:=", "",
	  "Version:=", "1.0",
	  "Notes:=", "",
	  "IconType:=", "",
	  "Owner:=", "Jane Doe",
	  "Email:=", "jdoe@email.com",
	  "Date:=", "3:46:31 PM Jul 26, 2020",</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "HasLabel:=", False,
      "IncludedParts:=", ["outer","teflon","inner","teflon_1"],
      "HiddenParts:=", [],
      "IncludedCS:=", [],
      "DefaultHandle:=", "Global",
      "IncludedParameters:=", ["bend_angle"],
      "ParameterDescription:=", ["bend_angle:=", ""]
    ],
    ["NAME:DesignData",
      "Excitations:=", ["1","2"]
    ],
    ["NAME:ImageFile",
      "ImageFile:=", ""
    ]
)
```

Section

Creates a 2D cross-section of the selection in the specified plane.

UI Access	**Modeler** > **Surface** > **Section**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<SelectionsArray>	Array	Structured array. See: SelectionsArray.

<Parameters>	Array	Structured array.
		`["NAME:SectionToParameters",`
		`"CreateNewObjects:=" , <boolean>,`
		`"SectionPlane:=" , <string "XY", "YZ", or "ZX">,`
		`"SectionCrossObject:=" , <boolean>]`

Return Value

None.

Python Syntax

Section (*<SelectionsArray>*, *<Parameters>*)

Python Example

```
oEditor.Section(
["NAME:Selections",
  "Selections:="          , "Cone1",
  "NewPartsModelFlag:="   , "Model"
],
["NAME:SectionToParameters",
  "CreateNewObjects:="    , True,
  "SectionPlane:="        , "XY",
  "SectionCrossObject:="  , False
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SeparateBody

Separates the bodies of specified multi-lump objects.

UI Access	**Modeler** > **Boolean** > **Separate Bodies**.
Parameters	Name: *<SelectionsArray>* — Type: Array — Description: Structured array. See: SelectionsArray.
Return Value	None.

Python Syntax	SeparateBody (*<SelectionsArray>*)

Python Example	`oEditor.SeparateBody(`
	`["NAME:Selections",`
	`  "Selections:=", "Rectangle1,Circle1",`
	`  "NewPartsModelFlag:=", "Model"`
	`])`

SetModelUnits

Sets the model units.

UI Access	**Modeler** > **Units**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array.
	`["NAME:Units Parameter",`
	`"Units:=", <string>,`
	`"Rescale:=", <boolean True to rescale model; else False>]`
	To see valid unit strings, select **Modeler** > **Units**.</td></tr></table>

Return Value	None.

Python Syntax	SetModelUnits (<*Parameters*>)

Python Example	`oEditor.SetModelUnits(`
	`["NAME:Units Parameter",`
	`"Units:=", "km",`
	`"Rescale:=", False`
	`])`

SetWCS

Sets the working coordinate system.

UI Access	**Modeler** > **Coordinate System** > **Set Working CS**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:SetWCS Parameter", "Working Coordinate System:=", <string CS name>, "RegionDepCSOk:=" , <boolean True if region-dependent; else False]
Return Value	None.		

Python Syntax	SetWCS (<*Parameters*>)
Python Example	oEditor.SetWCS(["NAME:SetWCS Parameter", "Working Coordinate System:=", "Global", "RegionDepCSOk:=", False])

ShowWindow

Opens the active 3D Modeler window.

UI Access	None.
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ShowWindow
Python Example	`oEditor.ShowWindow()`

Simplify

Converts a complex MCAD object into simpler primitives which are easy to mesh and solve.

UI Access	**Modeler** > **Model Preparation** > **Simplify**.

Parameters

Name	Type	Description
<Selections>	Array	Structured array. See: SelectionsArray.

<Parameters>	Array	Structured array.
		`["NAME:SimplifyParameters",`
		`"Type:=", <string fit type>,`
		`"ExtrusionAxis:=", <string>,`
		`"Cleanup:=", <boolean>,`
		`"Splitting:=", <boolean>,`
		`"SeparateBodies:=", <boolean>,`
		`"CloneBody:=", <boolean>,`
		`"Generate Primitive History:=", <boolean>,`
		`"NumberPointsCurve:=", <integer>,`
		`"LengthThresholdCurve:=", <integer>]`

<CreateGroup>	Array	Structured array. `["CreateGroupsForNewObjects:=",<boolean>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	Simplify (<*Selections*>, <*Parameters*>, <*CreateGroup*>)

Python Example	`oEditor.Simplify(`
	`["NAME:Selections",`
	`"Selections:=", "Cylinder1,Box2",`
	`"NewPartsModelFlag:=", "Model"],`
	`["NAME:SimplifyParameters",`
	`"Type:=", "Polygon Fit",`
	`"ExtrusionAxis:=", "Auto",`
	`"Cleanup:=", True,`
	`"Splitting:=", True,`
	`"SeparateBodies:=", False,`
	`"CloneBody:=", False,`
	`"Generate Primitive History:=", True,`
	`"NumberPointsCurve:=", 3,`
	`"LengthThresholdCurve:=", 20],`
	`["CreateGroupsForNewObjects:=", True]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Split

Splits the specified object(s) along a plane.

<table>
<tr><td>UI Access</td><td>Modeler > Boolean > Split.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:SplitToParameters",
  "SplitPlane:=" , <string "XY", "YZ", "ZX", or "Dummy">,
  "WhichSide:=" , <string "PositiveOnly", "NegativeOnly" or "Both">,
  "ToolType:=" , <string "PlaneTool" or "FaceTool">,
  "ToolEntityID:=" , <-1 if using SplitPlane; else FaceID>,
  "SplitCrossingObjectsOnly:=", <boolean>,
  "DeleteInvalidObjects:=", <boolean>]</pre>
You can split an object either along an existing plane , or by creating a plane using an edge.
For existing plane:
• <code>SplitPlane</code> is "XY", "YZ", or "ZX"
• <code>ToolType</code> is "PlaneTool"
• <code>ToolEntityID</code> is -1</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			For an edge: • `SplitPlane` is "Dummy" • `ToolType` is "EdgeTool" • `ToolEntityID` is the face ID
Return Value	None.		

Python Syntax	Split(<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.Split(`
	`["NAME:Selections",`
	`  "Selections:="  , "Box1",`
	`  "NewPartsModelFlag:="  , "Model"`
	`],`
	`["NAME:SplitToParameters",`
	`  "SplitPlane:="  , "XY",`
	`  "WhichSide:="  , "PositiveOnly",`
	`  "ToolType:="  , "PlaneTool",`
	`  "ToolEntityID:="  , -1,`
	`  "SplitCrossingObjectsOnly:=", False,`
	`  "DeleteInvalidObjects:=", True`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Stitch

Stitches selected sheets.

UI Access	**Modeler** > **Model Preparation** > **Stitch Sheets**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Selections*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. `Array("NAME:StitchParameters",` `  "MaxTol:=", <integer maximum tolerance>)`</td></tr></table>
Return Value	None.

Python Syntax	Stitch (<*Selections*>, <*Parameters*>)

Python Example	`oEditor.Stitch(`
	`["NAME:Selections",`
	`  "Selections:=", "Rectangle3,Rectangle4",`
	`  "NewPartsModelFlag:=", "Model"],`
	`["NAME:StitchParameters",`
	`  "MaxTol:=", -1]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Subtract

Subtracts the specified object(s).

<table>
<tr><td>UI Access</td><td>Modeler > Boolean > Subtract.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Selections></td><td>Array</td><td>Structured array.
["NAME:Selections",
"Blank Parts:=" , <string object name(s)>,
"Tool Parts:=" , <string object name(s)>]
The Tool Parts object is the object being removed.
The Blank Parts object has any Tool Parts overlap removed.
Either string can contain more than one object.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:SubtractParameters",
"KeepOriginals:=" , <boolean>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	Subtract(<*Selections*>, <*Parameters*>)
Python Example	oEditor.Subtract(["NAME:Selections", "Blank Parts:=", "Rectangle1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Tool Parts:=", "Rectangle2"], ["NAME:SubtractParameters", "KeepOriginals:=", False])

SweepFacesAlongNormal

Sweep the specified face(s) along normal.

UI Access	**Modeler** > **Surface** > **Sweep Faces Along Normal**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr><tr><td><*parameters*></td><td>Array</td><td>Structured array. ["NAME:Parameters", "NAME:SweepFaceAlongNormalToParameters", "FacesToDetach:=", <faceIDarray>, "LengthOfSweep:=", "<value><units>"]</td></tr></table>
Return Value	None

Python Syntax	SweepFacesAlongNormal(<*SelectionsArray*> <*parameters*>)
Python Example	oEditor.SweepFacesAlongNormal(["NAME:Selections",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Selections:=", "Rectangle1", "NewPartsModelFlag:=", "Model"], ["NAME:Parameters", "NAME:SweepFaceAlongNormalToParameters", "FacesToDetach:=", [183], "LengthOfSweep:=", "0.1mm"])

ThickenSheet

Thickens a sheet object to convert it to a 3D object.

UI Access	**Modeler > Surface > Thicken Sheet**		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.
	<parameters>	Array	Structured array. ["NAME:SheetThickenParameters", "Thickness:=" , <string>, "BothSides:=" , <boolean>]
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ThickenSheet (<*SelectionsArray*> <*parameters*>)
Python Example	oEditor.ThickenSheet(["NAME:Selections", "Selections:=" , "Rectangle3", "NewPartsModelFlag:=" , "Model"], ["NAME:SheetThickenParameters", "Thickness:=" , "0.01mm", "BothSides:=" , False])

UncoverFaces

Uncovers the specified face(s).

UI Access	**Modeler** > **Surface** > **Uncover Faces**		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
	<*parameters*>	Array	Structured array. ["NAME:Parameters", ["NAME:UncoverFacesParameters", "FacesToUncover:=" , <array of face IDs>]]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	UncoverFaces(<*SelectionsArray*> <*parameters*>)

Python Example	`oEditor.UncoverFaces(`
	`["NAME:Selections",`
	`  "Selections:="  , "Box1",`
	`  "NewPartsModelFlag:="  , "Model"`
	`],`
	`["NAME:Parameters",`
	`  ["NAME:UncoverFacesParameters",`
	`  "FacesToUncover:="  , [12,16,18]`
	`  ]`
	`])`

Unite

Unites the specified objects.

UI Access	**Modeler** > **Boolean** > **Unite**		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<parameters>	Array	Structured array.
			`["NAME:UniteParameters",`
			`"KeepOriginals:=" , <boolean>,`
			`"TurnOnNBodyBoolean:=", <boolean>]`
			Note:
			TurnOnNBodyBoolean is set to True by default. Enabled by Parasolid, this unites *n* bodies in a single step, rather than uniting them one by one.

Return Value	None.		

Python Syntax	Unite(<*SelectionsArray*>, <*parameters*>)

Python Example	`oEditor.Unite(`
	`["NAME:Selections",`
	`"Selections:=", "Rectangle1,Rectangle2"`
	`],`
	`["NAME:UniteParameters",`
	`"KeepOriginals:=", False,`
	`"TurnOnNBodyBoolean:=", True`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Ungroup

Ungroups a specified history tree group.

UI Access	**Modeler > Group > Ungroup**		
Parameters	Name	Type	Description
	<Groups>	Array	Structured array. `["Groups:=", <array of group names to ungroup>]`
Return Value	None		

Python Syntax	Ungroup(*<Groups>*)
Python Example	`oEditor.Ungroup(["Groups:=", ["Group1"]] )`

WrapSheet

Wraps a sheet object to another object.

UI Access	None.		
Parameters	Name	Type	Description
	<SelectionsArray>	Array	Structured array. See: SelectionsArray.

	<Parameters>	Array	Structured array.
			`["NAME:WrapSheetParameters",`
			`  "Imprinted:=", <boolean>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	WrapSheet (<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.WrapSheet(`
	`["NAME:Selections",`
	`  "Selections:=","Rectangle1,Box1"`
	`],`
	`["NAME:WrapSheetParameters",`
	`  "Imprinted:=", True`
	`]`
	`)`

Other oEditor Commands

AddDefinitionFromBlock

AddDefinitionFromLibFile

AddViewOrientation

BreakUDMConnection

ChangeProperty

Delete

FitAll

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GeometryCheckAndAutofix

GetBodyNamesByPosition

GetChildNames [Modeler]

GetChildOjbect [Modeler]

GetChildTypes [Modeler]

GetEdgeByPosition

GetEdgeIDsFromObject

GetEdgeIDsFromFace

GetEntityListIDByName

GetExtendedDefinitionObject

GetFaceArea

GetFaceByPosition

GetFaceCenter

GetFaceIDs

GetGeometryModelerMode

GetMatchedObjectName

GetModelBoundingBox

GetModelUnits

GetNumObjects

GetObjectAttributeValues

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetObjectIDByName

GetObjectName

GetObjectNameByFaceID

GetObjectsByMaterial

GetObjectsInGroup

GetObjectVolume

GetPropertyValue

GetPropEvaluatedValue

GetPropNames

GetPropSIValue

GetPropValue

GetSelections

GetUserPosition

GetVertexIDsFromEdge

GetVertexIDsFromFace

GetVertexIDsFromObject

GetVertexPosition

PageSetup

RenamePart

SetObjectAttributeValues

SetPropValue [Modeler]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Setting User-Defined Attributes

AddDefinitionFromLibFile

Adds a material definition from a library file (e.g. AMAT file), by name and library type (using definition folder name) . This scripting command directly supports the .AMAT (or .ASURF) definition formats.

UI Access	N/A
Parameters	

Name	Type	Description
<*FilePath*>	String	Path of the library file (i.e. AMAT file or ASURF file)
<*defName*>	String	Which definition to use, required because a lib file can have multiple definitions.
<*defFolderName*>	String	Library type (by definition folder name).
<*newTimeStamp*>	String	New timestamp string (time_t as integer, number of seconds since 1/1/1970 12:00am), default is current time
<*replaceExisting*>	Boolean	True to replace existing, False to choose a new unique name if an existing definition is found, default is False

Return Value	Property scripting object for the definition.
Python Syntax	AddDefinitionFromLibFile(<*FilePath*>, <*defName*>, <*defFolderName*>, <*newTimeStamp*>,<*replaceExisting*>)

Python Example	`oProject = oDesktop.NewProject()`
	`oProject.InsertDesign("HFSS", "HFSSDesign1", "DrivenModal", "")`
	`oDesign = oProject.SetActiveDesign("HFSSDesign1")`
	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oEditor.CreateBox(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ["NAME:BoxParameters",
    "XPosition:=" , "-0.4mm",
    "YPosition:=" , "-1mm",
    "ZPosition:=" , "0mm",
    "XSize:=" , "1.4mm",
    "YSize:=" , "1.6mm",
    "ZSize:=" , "0.6mm"
  ],
  ["NAME:Attributes",
    "Name:=" , "Box1",
    "Flags:=" , "",
    "Color:=" , "(143 175 143)",
    "Transparency:=" , 0,
    "PartCoordinateSystem:=", "Global",
    "UDMId:=" , "",
    "MaterialValue:=" , "\"vacuum\"",
    "SurfaceMaterialValue:=", "\"\"",
    "SolveInside:=" , True, "ShellElement:=" , False,
    "ShellElementThickness:=", "0mm",
    "IsMaterialEditable:=" , True,
    "UseMaterialAppearance:=", False,
    "IsLightweight:=" , False
  ])
oDefinitionManager = oProject.GetDefinitionManager()
scriptDir = os.path.dirname(os.path.realpath(__file__))
amatFilePath = scriptDir + '/material0.txt'
materialName = "material0"
materialNameInQuotes = "\"" + materialName + "\""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
added = oDefinitionManager.AddDefinitionFromLibFile(amatFilePath, mater-
ialName, "Materials", "")
addedName = ''
  if isinstance(added, basestring):
addedName = added
  elif isinstance(added, list):
addedName = added[0]
else:
  addedName = added.GetName().replace("Materials:", "")
AddInfoMessage(os.path.basename(__file__) + " result: " + addedName)
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers",
        "Box1"
      ],
      ["NAME:ChangedProps",
      ["NAME:Material",
        "Value:=", materialNameInQuotes
      ]
    ]
  ]
])
```

AssignSurfaceMaterial

Assigns a material to specified surfaces.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SelectionsArray></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr>
<tr><td><AttributesArray></td><td>Array</td><td>Structured array. See: AttributesArray.

This script supports the following attributes:
MaterialValue
SolveInside
ShellElement
ShellElementThickness
IsMaterialEditable
UseMaterialAppearance
IsLightweight</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	AssignSurfaceMaterial(<*SelectionsArray*>, <*AttributesArray*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oEditor.AssignSurfaceMaterial(["NAME:Selections", "AllowRegionDependentPartSelectionForPMLCreation:=", True, "AllowRegionSelectionForPMLCreation:=", True, "Selections:=" , "Rectangle1"], ["NAME:Attributes", "MaterialValue:=" , "diamond", "SolveInside:=" , False, "ShellElement:=" , False, "ShellElementThickness:=" , "nan", "IsMaterialEditable:=" , True, "UseMaterialAppearance:=" , False, "IsLightweight:=" , False])

BreakUDMConnection

Breaks a current UDM connection to Discovery.

UI Access	Break Connection.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SelectionsArray*></td><td>Array</td><td>Structured array. See: SelectionsArray.</td></tr></table>
Return Value	None.

Python Syntax	BreakUDMConnection (<*SelectionsArray*>)

Python Example	`oEditor.BreakUDMConnection(`
	`[`
	`"NAME:Selections",`
	`"Selections:=" , "Discovery1"`
	`])`

ChangeProperty

Changes the properties of an object in the history tree.

UI Access	Right-click an object in the History Tree and select **Properties**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*propertyArgs*></td><td>Array</td><td>Structured array. The properties vary depending on the object. Due to the number of potential configurations, it is recommended that you generate this script using the UI's **Automation** tab.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	**Example: Changing the Position of a Box**
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DCmdTab",`
	`  ["NAME:PropServers" , "Box1:CreateBox:1" ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Position", "X:=" , "0.35in", "Y:=" , "0.55in", "Z:=" , "0in"`
	`  ]`
	`  ]`
	`  ]`
	`  ])`
	
	Example: Offset the Position of a 3D Component or Layout Component with a Variable
	`oDesign.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:LocalVariableTab",`
	`  ["NAME:PropServers",`
	`  "LocalVariables"`
	`  ],`
	`  ["NAME:NewProps",`
	`  ["NAME:zH",`
	`  "PropType:=" , "VariableProp",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "UserDef:=" , True,
          "Value:=" , "50mm"
          ]
        ]
      ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers",
        "LC1_1"
      ],
      ["NAME:ChangedProps",
        ["NAME:Position",
        "X:=" , "0mm",
        "Y:=" , "0mm",
        "Z:=" , "zH"
        ]
      ]
    ]
  ])
```

Example: Changing a Box's Material and Wireframe Display

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers" , "Box1" ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ["NAME:ChangedProps",
        ["NAME:Material", "Value:=" , "\"vacuum\"" ],
        ["NAME:Display Wireframe", "Value:=" , True ]
      ]
    ]
  ])
```

Example: Change Default Handle (reference coordinate system) for a 3D Component or Layout Component

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers","LC1_1"],
      ["NAME:ChangedProps",
        ["NAME:Handle","Value:=" , " NotchOutY"]
      ]
    ]
  ])

oEditor.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:Maxwell2D",
       ["NAME:PropServers", "Design Settings" ],
      ["NAME:ChangedProps",
        ["NAME:Model settings/Depth", "Value:=" , "2.4mm" ]
       ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]])

CloseAllWindows[Editor]

Closes all windows belong to current 3D Modeler editor.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	CloseAllWindows()
Python Example	`oEditor.CloseAllWindows()`

Delete

Deletes the specified object(s).

UI Access	**Edit** > **Delete**.
Parameters	Name | Type | Description <*SelectionsArray*> | Array | Structured array. See SelectionsArray.
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Delete(<*SelectionsArray*>)

Python Example	`oEditor.Delete(`
	`  ["NAME:Selections",`
	`  "Selections:=", "Rectangle1,Rectangle2"`
	`])`

FitAll

Fits the design to the modeling area.

UI Access	**View** > **Fit All** > **All Views**.
Parameters	None.
Return Value	None.

Python Syntax	FitAll()
Python Example	`oEditor.FitAll()`

GenerateAllUserDefinedModels

Generates all user defined models.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	Array of models generated.

Python Syntax	GenerateAllUserDefinedModels ()
Python Example	`oEditor.GenerateAllUserDefinedModels()`

GenerateUserDefinedModel

Generates specified user defined model(s).

UI Access	N/A
Parameters	Name | Type | Description <*SelectionsArray*> | Array | Structured array. See: SelectionsArray.
Return Value	Array of models generated.

Python Syntax	GenerateUserDefinedModel (<*SelectionsArray*>)
Python Example	`oEditor.GenerateUserDefinedModel(["NAME:Selections", "Selections:=", "model1, model2"])`

GeometryCheckAndAutofix

Use: Runs Geometry Check and optionally applies autofixes.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: **HFSS 3D Layout > Geometry Check**

Syntax: GeometryCheckAndAutofix <ChecksArray>,

minimum_area_meters_squared,

<FixesArray>

Return Value: None

Parameters: **<ChecksArray> - Array("NAME:checks", <check 1>, <check 2>, …, <check n>)**

Specify the checks that should be included. Specifying fewer checks may speed up execution but may also result in less problems reported in the message manager (or the logfile) and consequently less problems that can be fixed by autofixes.

The following are valid checks that can be specified:

- "Self-Intersecting Polygons"
- "Disjoint Nets (Floating Nodes)"
- "DC-Short Errors"
- "Identical/Overlapping Vias"
- "Misaligments"

There may be no checks, all 5 of the checks, or anything in between. The order that checks are specified in is not relevant.

minimum_area_meters_squared

Specify a decimal value for the minimum area (e.g. .000015) optionally in scientific notation (e.g. 2E-006). Cutouts smaller than this minimum area may be ignored during validation check.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<FixesArray> - Array("NAME:fixes", <fix 1>, <fix 2>, …, <fix n>)

Specify the autofixes that should be applied if they are found by a check.

The following are valid fixes that can be specified:

- "Self-Intersecting Polygons"
- "Disjoint Nets",
- "Identical/Overlapping Vias"
- "Traces-Inside-Traces Errors"
- "Misalignments (Planes/Traces/Vias)"

There may be no fixes specified, all 5 fixes specified, or anything in between. The order that fixes are specified in is not relevant.

GetBodyNamesByPosition

Returns the names of objects that contact a specified point.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><positionParameters></td><td>Array</td><td>Structured array containing position coordinates for active coordinate system.
["NAME:Parameters",
"XPosition:=", <string>,
"YPosition:=", <string>,
"ZPosition:=", <string>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Array containing string object names.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetBodyNamesByPosition (<*positionParameters*>)

Python Example	`oEditor.GetBodyNamesByPosition(`
	`  ["NAME:Parameters",`
	`  "XPosition:=","0mm",`
	`  "YPosition:=","15mm",`
	`  "ZPosition:=","0mm"`
	`  ]`
	`)`

GetChildNames [Modeler]

Returns the names of children for a specified input. For 3D Components and UDMs, these commands do not return parts, coordinate systems, plans, as top-level modeler children.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A		
Parameters	Name	Type	Description
	<*category*>	String	*Optional*. Passing no input returns the list of possible strings: • "AllParts" – Returns the names of all parts. • "CoordinateSystems" – Returns the names of all

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			coordinate systems. • "Groups" – Returns the names of all groups. • "Lists" – Returns the names of all lists. • "ModelParts" – Returns names of model parts. • "NonModelParts" – Returns the names of non-model parts. • "Planes" – Returns the names of all planes. • "Points" – Returns the names of all points. • "SubmodelDefinitions" – Returns the names of sub-model definitions.
Return Value	Array containing object names in the selected category.		

Python Syntax	GetChildNames(<*category*>)

Python Example	Standalone Example:
	`oEditor.GetChildNames("ModelParts")`
	`oEditor.GetChildNames("Points")`
	Examples used in Conjunction with **GetChildObject** and **GetPropNames**:
	`oDesign = oProject.GetActiveDesign()`
	`oModel = oDesign.GetChildObject("3D Modeler")`
	`oModel.GetChildTypes()`
	• This returns an array containing strings: "ModelParts", "AllParts", "NonModelParts", "CoordinateSystems", "Points", "Planes", "SubmodelDefinitions", "Groups", and "Lists".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oModel.GetChildNames("ModelParts")`
	• This returns an array containing string model parts. For example: "WG_Interior", "WG", "MT_Interior", "MT", "Box2", "Cylinder1".
	`oWGi = oModel.GetChildObject("WG_Interior")`
	• This sets the object WG_Interior to variable oWGi.
	`oWGi.GetPropNames()`
	• This returns property names from child object assigned to oWGi. In this case: "Name", "Material", "Material/SIValue", "Material/EvaluatedValue", "Solve Inside", "Orientation", "Orientation/Choices", "Model", "Display Wireframe", "Material Appearance", "Color", "Color/Red", "Color/Green", "Color/Blue", and "Transparent".
	`oEditor = oDesign.SetActiveEditor("3DModeler")`
	`udm = oEditor.GetChildObject("Discovery1")`
	`udm.GetChildNames("Group")`
	• This returns `['Extruded', 'Spherical']`

GetChildObject [Modeler]

Returns a 3D modeler child object, which can be assigned to a variable. Will return normally if there are no active objects. For 3D Components and UDMs, these commands do not return parts, coordinate systems, or plans as top-level modeler children.

> **Note:** This command is not supported by the EMIT and Circuit design types.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*object*></td><td>String</td><td>In this case, "3D Modeler".</td></tr></table>
Return Value	Returns the 3D Modeler object. In the examples below, it is assigned to the variable oEditor.

Python Syntax	GetChildObject(<object>)

Python Example	**Standalone Example:**
	`oDesign = oProject.GetActiveDesign()`
	`oEditor = oDesign.GetChildObject("3D Modeler")`
	Example used in Conjunction with GetChildNames and GetPropNames:
	`oDesign = oProject.GetActiveDesign()`
	`oModel = oDesign.GetChildObject("3D Modeler")`
	`oModel.GetChildTypes()`
	• This returns an array containing strings: "ModelParts", "AllParts", "NonModelParts", "CoordinateSystems", "Points", "Planes", "SubmodelDefinitions", "Groups", and "Lists".
	`oModel.GetChildNames("ModelParts")`
	• This returns an array containing string model parts. For example: "WG_Interior", "WG", "MT_Interior", "MT", "Box2", "Cylinder1".
	`oWGi = oModel.GetChildObject("WG_Interior")`
	• This sets the object WG_Interior to variable oWGi.
	`oWGi.GetPropNames()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• This returns property names from child object assigned to oWGi. In this case: "Name", "Material", "Material/SIValue", "Material/EvaluatedValue", "Solve Inside", "Orientation", "Orientation/Choices", "Model", "Display Wireframe", "Material Appearance", "Color", "Color/Red", "Color/Green", "Color/Blue", and "Transparent".
	UDM Discovery Example with Hierarchy:
	`oDesign = oProject.GetActiveDesign()`
	`oEditor = oDesign.SetActiveEditor("3DModeler")`
	`udm = oEditor.GetChildObject("Discovery1")`
	`udm.GetChildNames("Group")`
	returns `['Extruded', 'Spherical']`

GetChildTypes [Modeler]

Gets child types of queried designs or editors obtained by GetActiveProject() and GetActiveDesign() commands. For 3D Components and UDMs, these commands do not return parts, coordinate systems, plans, as top-level modeler children.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	None.
Return Value	Array containing the names of child types.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetChildTypes()

Python Example	**Standalone Example:**
	`oDesign = oProject.GetActiveDesign()`
	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oEditor.GetChildTypes()`
	• This returns an array containing strings: "Part", "CoordinateSystem", "Group", "ComponentDefinition", "UserDefinedModel", "Point", "Plane", and "List".
	Note: Users can query the Part type to see if it is a model or nonmodel.
	Example Used in Conjunction with GetChildNames:
	`oProject = oDesktop.GetActiveProject()`
	`oDesign = oProject.GetActiveDesign()`
	`oDesign.GetChildNames()`
	• This returns an array containing names of child objects for the design. For example: "Boundaries", "Nets", "Analysis", "Optimetrics", "Radiation", "Results", and "3D Modeler".
	`oDesign.GetChildTypes()`
	• This returns an array containing the child types. For example: "Module", "Editor", and "Variable".

GetEdgeByPosition

Returns the ID for edge(s) that contact a specified point.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*positionParameters*></td><td>Array</td><td>Structured array.
	`["NAME:EdgeParameters",`
	`"BodyName:=", <string object name>,`
	`"Xposition:=", <string>,`
	`"YPosition:=", <string>,`
	`"ZPosition:=", <string>]`
	Note:
	For 2D XY Designs, ZPosition should be set to "0".
	For 2D RZ Designs, YPosition should be set to "0".</td></tr></table>

Return Value	Array containing string edge IDs.

Python Syntax	GetEdgeByPosition (<*positionParameters*>)

Python Example	`oEditor.GetEdgeByPosition(`
	`["NAME:EdgeParameters",`
	`"BodyName:=", "Box1",`
	`"Xposition:=", "10mm",`
	`"YPosition:=", "0mm",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ZPosition:=", "10mm"
  ]
)
```

GetEdgeIDsFromFace

Returns the edge IDs for a specified Face ID.

UI Access	N/A
Parameters	Name: *<faceID>* — Type: Integer — Description: ID of the specified face.
Return Value	Array containing string edge IDs.

Python Syntax	GetEdgeIDsFromFace (*<faceID>*)
Python Example	`oEditor.GetEdgeIDsFromFace(20)`

GetEdgeIDsFromObject

Returns the edge IDs for a specified object.

UI Access	N/A
Parameters	Name: *<object>* — Type: String — Description: Object name.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Array containing string edge IDs.

Python Syntax	GetEdgeIDsFromObject (<*object*>)
Python Example	`oEditor.GetEdgeIDsFromObject("Box1")`

GetEdgeLength

Returns the length of edges for a specified Edge ID.

UI Access	N/A
Parameters	Name | Type | Description <*edgeID*> | Integer | ID of the specified edge.
Return Value	Integer containing the length of found edges.

Python Syntax	GetEdgeLength (<*EdgeID*>)
Python Example	`oEditor.GetEdgeLength(20)`

GetEntityIDsContainedByNamedSelection

Returns the list of entity IDs contained in the named selection. See CreateNamedSelection.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description <*selectionName*> | String | selection name.
Return Value	String containing the list of entity IDs contained in the named selection.

Python Syntax	GetEntityIDsContainedByNamedSelection (<*selectionName*>)
Python Example	`entityIDs = oEditor.GetEntityIDsContainedByNamedSelection("FaceSelection1")`

GetEntityListIDByName

Returns the specified entity list's ID number. See CreateNamedSelection.

UI Access	N/A
Parameters	Name | Type | Description <*listName*> | String | List name.
Return Value	String containing the entity list ID number.

Python Syntax	GetEntityListIDByName (<*listName*>)
Python Example	`oEditor.GetEntityListIDByName("MyList")`

GetFaceArea

Returns the area of a specified face.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name | Type | Description <*faceID*> | Integer | ID of specified face.
Return Value	Long integer of face area.

Python Syntax	GetFaceArea (<*faceID*>)
Python Example	`oEditor.GetFaceArea(19)`

GetFaceCenter

Returns the center position of a specified face.

UI Access	N/A
Parameters	Name | Type | Description <*planarFaceID*> | Long | Face ID
Return Value	Array containing string X, Y, Z coordinates.

Python Syntax	GetFaceCenter (<*planarFaceID*>)
Python Example	`oEditor.GetFaceCenter(12)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetFaceByPosition

Returns the face ID located at a specified position.

> **Note:**
>
> The coordinates must point to exactly one face, not a vertex or edge where two or more faces join.

UI Access	N/A

Parameters	Name: *<FaceParameters>*
	Type: Array
	Description: Structured Array.
	`["NAME:FaceParameters",`
	`"BodyName:=", <string>,`
	`"XPosition:=", <string>,`
	`"YPosition:=", <string>,`
	`"ZPosition:=", <string>]`

Return Value	Integer Face ID.

Python Syntax	GetFaceByPosition(<*FaceParameters*>)

Python Example	`oEditor.GetFaceByPosition(`
	`["NAME:FaceParameters",`
	`"BodyName:=", "Box1",`
	`"XPosition:=", "0.2mm",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "YPosition:=", "-0.2mm",
   "ZPosition:=", "0.4mm"
])
```

GetFaceIDs

Returns the face IDs associated with a specified object.

UI Access	N/A
Parameters	Name: *<objectName>*, Type: String, Description: Object name.
Return Value	Array containing string face IDs.

Python Syntax	GetFaceIDs (<*objectName*>)
Python Example	`oEditor.GetFaceIDs('Box1')`

GetFaceIDsOfSheet

Returns the face IDs associated with a specified sheet object.

UI Access	N/A
Parameters	Name, Type, Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*sheetName*>	String	Name of specified sheet object.
Return Value	Array containing string face IDs.		

Python Syntax	GetFaceIDsOfSheet(<*sheetName*>)
Python Example	`oEditor.GetFaceIDsOfSheet('Sheet1')`

GetMatchedObjectName

Returns all object names containing the input text string.

UI Access	N/A		
Parameters	Name	Type	Description
	<*wildcardText*>	String	Text string present in object name(s).
Return Value	Array containing string object names.		

Python Syntax	GetMatchedObjectName (<*wildcardText*>)
Python Example	`oEditor.GetMatchedObjectName('Box*')` `oEditor.GetMatchedObjectName('?ox?')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetModelBoundingBox

Returns the bounding box of the current model using the global coordinate system as reference. The outputs are in the defined modeler units. Only modeled objects are considered for this computation (non-modeled objects are ignored).

UI Access	N/A
Parameters	None.
Return Value	Array containing string Xmin, Ymin, Zmin, Xmax, Ymax, and Zmax values of the bounding box in the global coordinate system

Python Syntax	GetModelBoundingBox()
Python Example	`oEditor.GetModelBoundingBox()`

GetModelUnits

Returns the model's unit of measure.

UI Access	N/A
Parameters	None.
Return Value	String containing unit of measure.

Python Syntax	GetModelUnits()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.GetModelUnits()`

GetNumObjects

Returns the number of objects in a design.

UI Access	N/A
Parameters	None.
Return Value	Integer number of objects.

Python Syntax	GetNumObjects()
Python Example	`oEditor.GetNumObjects()`

GetObjectIDByName

Given an object's name, returns its ID. IDs are used with CreateEntityList.

UI Access	N/A
Parameters	Name | Type | Description *<objectName>* | String | Object name.
Return Value	Integer object ID.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetObjectIDByName(<*objectName*>)
Python Example	`oEditor.GetObjectIDByName('Box1')`

GetObjectName

Returns an object's name from its specified base index (creation order).

UI Access	N/A
Parameters	Name | Type | Description <*index*> | Integer | Base index (where '0' is the first item created)
Return Value	String containing object name.

Python Syntax	GetObjectName(<*index*>)
Python Example	`oEditor.GetObjectName(3)`

GetObjectNameByEdgeID

Returns an object name given an edge ID.

UI Access	N/A
Parameters	Name | Type | Description <*EdgeID*> | Integer | The edge ID.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	String containing object name.

Python Syntax	GetObjectNameByEdgeID(<*EdgeID*>)
Python Example	`oEditor.GetObjectNameByEdgeID(88)`

GetObjectNameByFaceID

Returns an object name given a face ID.

UI Access	N/A
Parameters	Name: Type: Integer; Description: The face ID. (<*FaceID*>)
Return Value	String containing object name.

Python Syntax	GetObjectNameByFaceID (<*FaceID*>)
Python Example	`oEditor.GetObjectNameByFaceID(88)`

GetObjectNameByID

Returns an object name given an ID.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description <*ObjID*> | Integer | The object ID.
Return Value	String containing object name.

Python Syntax	GetObjectNameByID(<*ObjID*>)
Python Example	`oEditor.GetObjectNameByID(88)`

GetObjectNameByVertexID

Returns an object name given a vertex ID.

UI Access	N/A
Parameters	Name | Type | Description <*VertexID*> | Integer | The vertex ID.
Return Value	String containing object name.

Python Syntax	GetObjectNameByVertexID(<*VertexID*>)
Python Example	`oEditor.GetObjectNameByVertexID(88)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetObjectsByMaterial

Returns a list of objects of a specified material.

UI Access	N/A
Parameters	Name | Type | Description *<material>* | String | Material name.
Return Value	Array containing string object names.

Python Syntax	GetObjectsByMaterial (<*material*>)
Python Example	`oEditor.GetObjectsByMaterial('copper')`

GetObjectShapeType

Returns the shape type of a specified object.

UI Access	N/A
Parameters	Name | Type | Description *<objName>* | String | Name of specified object. <csName> | String | Name of coordinate system of the object.
Return Value	String containing shape type.

Python Syntax	GetObjectShapeType(<*objName*>, <*csName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.GetObjectShapeType("box1", "Global")`

GetObjectsInGroup

Returns a list of objects in a specified group.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*group*></td><td>String</td><td>Group name. One of: "<*materialName*>" "<*assignmentName*>" "Non Model" "Solids" "Unclassified" "Sheets" "Lines" </td></tr></table>
Return Value	Array containing string object names.

Python Syntax	GetObjectsInGroup (<*group*>)
Python Example	`oEditor.GetObjectsInGroup('Sheets')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetObjectVolume

Returns an object's volume from its name).

UI Access	N/A
Parameters	Name | Type | Description *<name>* | String | Object name
Return Value	.Real

Python Syntax	GetObjectVolume(<*name*>)
Python Example	`oEditor.GetObjectVolume("Box1")`

GetObjPath [Editor]

Obtains the path to the 3D modeler.

UI Access	N/A
Parameters	None.
Return Value	String containing the path.

Python Syntax	GetObjPath()
Python Example	`oEditor.GetObjPath()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPartsForUserDefinedModel

Obtains parts from a specified user defined model.

UI Access	N/A
Parameters	Name | Type | Description <*udmName*> | String | Name of user defined model.
Return Value	Array of strings containing associated parts.

Python Syntax	GetPartsForUserDefinedModel (<*udmName*>)
Python Example	`oEditor.GetPartsForUserDefinedModel("OnDieSpiralInductor1")`

GetPoints [3D Modeler Editor]

Returns all the points defined in current 3D modeler.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing names of points.

Python Syntax	GetPoints ()
Python Example	`oEditor.GetPoints()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	Name | Type | Description <*PropName*> | String | Name of the property.
Return Value	String value of the evaluated value.

Python Syntax	GetPropEvaluatedValue (<*PropName*>)
Python Example	`oVar = oDesign.GetChildObject(" Variables/var")` `oVar.GetPropEvaluatedValue()`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Tip: Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

<table>
<tr><td>UI Access</td><td colspan="3">N/A</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PropTab></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:
PassedParameterTab ("Parameter Values")
DefinitionParameterTab (Parameter Defaults")
LocalVariableTab ("Variables" or "Local Variables")
ProjectVariableTab ("Project variables")
ConstantsTab ("Constants")
BaseElementTab ("Symbol" or "Footprint")
ComponentTab ("General")
Component("Component")
CustomTab ("Intrinsic Variables")
Quantities ("Quantities")
Signals ("Signals")</td></tr>
<tr><td><PropServer></td><td>String</td><td>An object identifier, generally returned from another script method, such as <code>Comp‡nst@R;2;3</code></td></tr>
<tr><td><PropName></td><td>String</td><td>Name of the property.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">String value of the property.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`
	`...`

GetPropNames [Modeler]

Returns the property names for the active model object or specified property values.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*IncludeReadOnly*></td><td>Boolean</td><td>Optional. • **True** - includes all properties. • **False** - returns only property names that can be changed. True by default.</td></tr></table>
Return Value	Returns property names of the current 3D Model object; includes the following: ["Name", "Material", "Solve Inside", "Orientation", "Orientation/Choices", "Model", "Group", "Display Wireframe", "Material Appearance", "Color", "Color/Red", "Color/Green", "Color/Blue", "Transparent"]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetPropNames(<*IncludeReadOnly*>)
Python Example	`oEditor.GetPropNames(ObjectName)`

GetPropSIValue

Returns the SI-Value for Value-Property and Variable. Return NAN for other property type if its value is not able to convert to be a double-floating point value.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PropName*>	String	Name of the property.
Return Value	Property value as a double floating value, or NAN if the property value cannot be converted to double floating point.		

Python Syntax	GetPropSIValue (<*PropName*>)
Python Example	`oCreateBox = oDesign.GetChildObject("3D Modeler/Box1/CreateBox:1")` `oCreateBox.GetPropValue("xSize")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	return "length / 2"
	`oCreateBox.GetPropEvaluatedValue("xSize")`
	return '0.4mm'
	`oCreateBox.GetPropSIValue("xSize")`
	return 0.0004

GetPropValue [Modeler]

Returns the property value for the active model object, or specified property values.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	Name | Type | Description *<propPath>* | | Optional. Child object's property path. See: Object Property Function Summary.
Return Value	The property value.

Python Syntax	GetPropValue(*<propPath>*)

Python Example	`Rh1 = oDesign.GetChildObject('Box1')`
	`Rh1.GetPropValue('Orientation')`
	Returns the specified Property Value:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`'Global'`

GetRelativeCoordinateSystems

Returns the relative coordinate systems in the current 3D modeler.

UI Access	N/A
Parameters	None.
Return Value	Array containing name of relative coordinate systems.

Python Syntax	GetRelativeCoordinateSystems()
Python Example	`oEditor.GetRelativeCoordinateSystems()`

GetSelections [Model Editor]

Returns an array of currently selected objects.

UI Access	N/A
Parameters	None.
Return Value	Array containing object IDs

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetSelections()
Python Example	`oEditor.GetSelections()`

GetSubGroupsInGroup

Returns subgroup names in a specified group.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*group*></td><td>String</td><td>Group name. One of: "<*materialName*>" "<*assignmentName*>" "Non Model" "Solids" "Unclassified" "Sheets" "Lines" </td></tr></table>
Return Value	Array containing subgroup names.

Python Syntax	GetSubGroupsInGroup(<*group*>)
Python Example	`oEditor.GetSubGroupsInGroup('Sheets')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetUserPosition

Returns a user's current coordinates in the 3D Modeler window.

UI Access	N/A
Parameters	Name: *<prompt>*, Type: String, Description: "Enter a point." Then click a point in the 3D Modeler window.
Return Value	Array containing X, Y, Z coordinates.

Python Syntax	GetUserPosition(*<prompt>*)
Python Example	`oEditor.GetUserPosition("Enter a point.")`

GetVertexIDFromNameForFirstOperation

Returns vertex ID associated with a specified vertex belongs to the first operation of a part.

UI Access	N/A
Parameters	Name: *<PartName>*, Type: String, Description: Name of specified part. Name: *<VertexName>*, Type: String, Description: Name of specified vertex.
Return Value	Integer representing vertex ID.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetVertexIDFromNameForFirstOperation(<*PartName*>, <*VertexName*>)
Python Example	`oEditor.GetVertexIDFromNameForFirstOperation("Part1", "Vertex1")`

GetVertexIDsFromEdge

Returns vertex IDs associated with a specified edge.

UI Access	N/A
Parameters	Name | Type | Description <*edgeID*> | Integer | Edge ID.
Return Value	Array containing string vertex IDs.

Python Syntax	GetVertexIDsFromEdge(<*edgeID*>)
Python Example	`oEditor.GetVertexIDsFromEdge(10)`

GetVertexIDsFromFace

Returns vertex IDs associated with a specified face.

UI Access	N/A
Parameters	Name | Type | Description <*faceID*> | Integer | Face ID.
Return Value	Array containing string vertex IDs.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetVertexIDsFromFace(<*faceID*>)
Python Example	`oEditor.GetVertexIDsFromFace(10)`

GetVertexIDsFromObject

Returns vertex IDs associated with a specified object.

UI Access	N/A
Parameters	Name | Type | Description <*objectName*> | String | Object name.
Return Value	Array containing string vertex IDs.

Python Syntax	GetVertexIDsFromObject(<*objectName*>)
Python Example	`oEditor.GetVertexIDsFromObject('Box1')`

GetVertexPosition

Returns an array of coordinates for a specified vertex.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name: *<vertexID>* — Type: Integer — Description: Vertex ID.
Return Value	Array containing X, Y, Z coordinates.

Python Syntax	GetVertexPosition(<*vertexID*>)
Python Example	`oEditor.GetVertexPosition(1)`

GetWireBodyNames

Returns the wire body names in current 3D modeler.

UI Access	N/A
Parameters	None.
Return Value	Array containing string of names.

Python Syntax	GetWireBodyNames()
Python Example	`oEditor.GetWireBodyNames()`

PageSetup

Specifies page settings for printing.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>File > Page Setup.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
[
"NAME:PageSetupData",
"margins:=", <SetupArray>
]</td></tr>
<tr><td><SetupArray></td><td>Array</td><td>Structured Array
[
"left:=", <string>,
"right:=", <string>,
"top:=", <string>,
"bottom:=", <string>)
]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	PageSetup(<*Parameters*>)

Python Example	`oEditor.PageSetup([`
	`"NAME:PageSetupData",`
	`"margins:=",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["left:=", "10mm", "right:=", "10mm", "top:=", "10mm", "bottom:=", "10mm"]])

RemoveBadEdges

Removes bad edges from specified list.

UI Access	N/A
Parameters	Name: *<EdgeList>* — Type: Array — Description: Array containing edge IDs.
Return Value	None.

Python Syntax	RemoveBadEdges (*<EdgeList>*)
Python Example	`oEditor.RemoveBadEdges([18,30])`

RemoveBadFaces

Removes bad faces from specified list.

UI Access	N/A
Parameters	Name: *<FaceList>* — Type: Array — Description: Array containing face IDs.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	RemoveBadFaces (<*FaceList*>)
Python Example	`oEditor.RemoveBadFaces([328,333])`

RemoveBadVertices

Removes bad vertices from specified list.

UI Access	N/A		
Parameters	Name	Type	Description
	<*VertexList*>	Array	Array containing vertex IDs.
Return Value	None.		

Python Syntax	RemoveBadVertices (<*VertexList*>)
Python Example	`oEditor.RemoveBadVertices([324,325])`

RenamePart

Renames an object.

UI Access	Enter new name in **Name** field.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><renameParametersArray></td><td>Array</td><td>Structured array. ["NAME:Rename Data", "Old Name:=", <string>, "New Name:=", <string>]</td></tr></table>
Return Value	None.

Python Syntax	oEditor.RenamePart(<*renameParametersArray*>)
Python Example	oEditor.RenamePart(['NAME:Rename Data', 'Old Name:=', 'partname', 'New Name:=', 'newpartname',])

SetPropValue [Modeler]

Sets the property value for the active model property.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

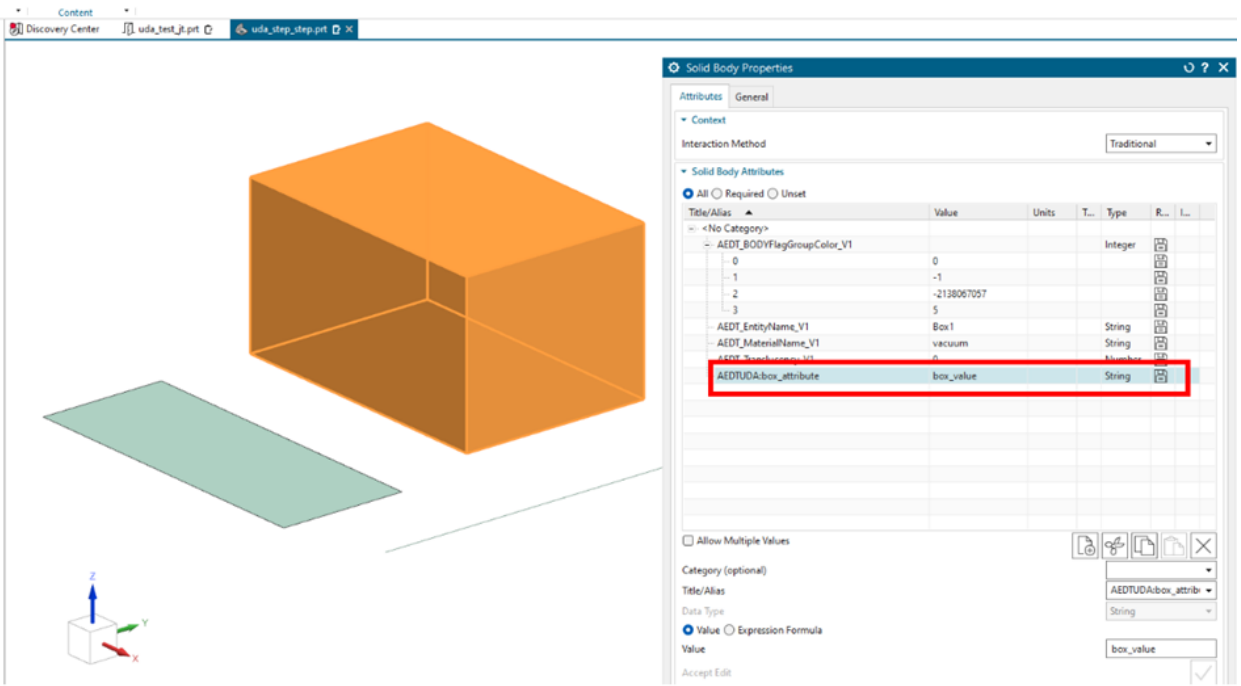
UI Access	Edit **Properties** on History Tree objects
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*propPath*></td><td>String</td><td>Child object's property path. See: Object Property Function Summary.</td></tr><tr><td><*value*></td><td>Varies</td><td>New property value.</td></tr></table>
Return Value	Boolean: • **True** – property found. • **False** – property not found.

Python Syntax	SetPropValue(<*propPath*>, <*value*>)
Python Example	`oEditor.SetPropValue("Color/r",111)`

Setting User-Defined Attributes

In Modeler, users can set arbitrary key-value pair strings on objects with a script command (this functionality is not available in the software interface). These user-defined attributes will be preserved when the model is exported as a JT or Step file for use in third-party CAD files, such as NX:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



They will also be recognized if the model is reimported into Ansys Electronics Desktop.

> **Note:** JT export does not support line body export. So an imported model will lose the line bodies, but user-defined attributes are maintained on exported object and sheet bodies.

The commands related to this functionality are listed below:

- SetObjectAttributeValues
- GetObjectAttributeValues

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- RemoveObjectAttributeValues

SetObjectAttributeValues

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*objectName*></td><td>String</td><td>Object to apply the attribute to; it can be a 3D object, a sheet or a line.</td></tr><tr><td><*Type*></td><td>String</td><td>Only possible value is "AEDTUDA"</td></tr><tr><td><*AttributePairs*></td><td>Array</td><td>List of key name and key value pairs</td></tr></table>
Return Value	Returns a value of 1 if attribute is successfully set

Python Syntax	SetObjectAttributeValues(<*objectName*>, "AEDTUDA", [“keyName”, “KeyValue”,..])

Python Example	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oEditor.SetObjectAttributeValues("Box1", "AEDTUDA", ["box_attribute", "box_value"])`
	`oEditor.SetObjectAttributeValues("Rectangle1", "AEDTUDA", ["sheet_attribute", "sheet_value"])`
	`oEditor.SetObjectAttributeValues("Line1", "AEDTUDA", ["line_attribute", "line_value"])`

GetObjectAttributeValues

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*objectName*></td><td>String</td><td>Name of object to query</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*Type*>	String	Only possible value is "AEDTUDA"
	<*stringFilter*>	String	Allows you to filter for specific UDAs and values if multiple UDAs are assigned to an object. Leave blank to query for all UDAs assigned to an object
Return Value	List of key value pairs		

Python Syntax	GetObjectAttributeValues(<*objectName*>, "AEDTUDA", <*stringFilter*>)

Python Example	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oEditor.GetObjectAttributeValues("Box1", "AEDTUDA", [" "])`
	Returns `["box_attribute", "box_value"]`

RemoveObjectAttributeValues

UI Access	None		
Parameters	Name	Type	Description
	<*objectName*>	String	Name of object that has the UDAs you want to remove
	<*Type*>	String	Only possible value is "AEDTUDA"
	<*stringFilter*>	String	Allows you to filter for specific UDAs and values if multiple UDAs are assigned to an object. Leave blank to delete all UDAs assigned to an object
Return Value	Returns a value of 1 if attribute is successfully deleted		

Python Syntax	RemoveObjectAttributeValues(<*objectName*>, "AEDTUDA", <*stringFilter*>)
Python Example	`oEditor = oDesign.SetActiveEditor("3D Modeler")` `oEditor.RemoveObjectAttributeValues("Box1", "AEDTUDA", [" "])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UDAs are available and editable when the component is open for editing. The UDAs of the component parts cannot be accessed or queried from the project/design level.

To edit a component, make sure to call it first:

```
oEditor = oDesign.SetActiveEditor("3D Modeler")

oProject.GetName()

'uda_comp'

oEditor.GetObjectAttributeValues("Box1", "AEDTUDA", [" "])
```

SetTopDownViewDirectionForActiveView

Sets active view to top-down view direction.

UI Access	N/A
Parameters	Name | Type | Description <*RelativeCS*> | String | Name of relative coordinate system
Return Value	None.

Python Syntax	SetTopDownViewDirectionForActiveView (<*RelativeCS*>)
Python Example	`oEditor.SetTopDownViewDirectionForActiveView("Global")`

SetTopDownViewDirectionForAllViews

Sets all views to top-down view direction.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name | Type | Description <*RelativeCS*> | String | Name of relative coordinate system
Return Value	None.

Python Syntax	SetTopDownViewDirectionForAllViews (<*RelativeCS*>)
Python Example	`oEditor.SetTopDownViewDirectionForAllViews("Global")`

UpdatePriorityList

Updates specified priority lists.

UI Access	N/A
Parameters	Name | Type | Description <*Lists*> | Array | Array of priority list names.
Return Value	None.

Python Syntax	UpdatePriorityList (<*Lists*>)
Python Example	`oEditor.UpdatePriorityList(["PriorityList1", "PriorityList2"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpgradeVersion

Upgrades legacy geometry to current version.

<table>
<tr><td>UI Access</td><td>Right-click on an operation icon in the history tree, select Upgrade Version.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Selections></td><td>Array</td><td>Structured array.
<pre>["NAME:Parameters",
  ["NAME:PartOperations",
    ["NAME:<string>",],
    "OperationIndices:=", <array of integers>
  ],
  ["NAME:UDMOperations"]
]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	UpgradeVersion (<*Selections*>)

Python Example	<pre>oEditor.UpgradeVersion([
	        "NAME:Parameters",
	        ["NAME:PartOperations",
	                ["NAME:source",
	                "OperationIndices:=", [0]]
	        ],</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:UDMOperations"]])

Validate3DComponent

Validates a 3D component.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*componentPath*></td><td>String</td><td>Path to a 3D component.</td></tr><tr><td><password></td><td>String</td><td>Optional. Password to access the component.</td></tr></table>
Return Value	Boolean: • **1** - component is valid. • **0** - component is not valid.

Python Syntax	Validate3DComponent (<*componentPath*>, <*password*>)
Python Example	`oEditor.Validate3DComponent(` `"C:/temp/component.3dcomp","")`

WriteHistoryTreeLayoutForTest

Writes history tree layout to a file.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*filePath*>	String	Path to the file.
<*OrgByMaterial*>	Integer	• **1** - organize objects by material. • **0** - do not organize by material.
<*OrgByAssignment*>	Integer	• **1** - organize sheets by assignment. • **0** - do not organize by assignment.
<*OrgByCompDefinition*>	Integer	• **1** - organize components by definition. • **0** - do not organize by definition.
<*DonotOrgUnderGroup*>	Integer	• **1** - do not organize within group. • **0** - organize within group.
<*ShowGroup*>	Integer	Optional. • **1** - show group. • **0** - do not show.

Return Value

None.

Python Syntax

WriteHistoryTreeLayoutForTest (<*filePath*>, <*OrgByMaterial*>, <*OrgByAssignment*>, <*OrgByCompDefinition*>, <*DonotOrgUnderGroup*>, <*ShowGroup*>)

Python Example

```
oEditor.WriteHistoryTreeLayoutForTest
("C:\temp", 1, 0, 0, 0, 1)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Cable Modeling Commands

Cables are created and edited as design objects using the Cable Seutp module, and then inserted as native components. For example, elements of a cable are created:

```
oModule = oDesign.GetModule("CableSetup")

oModule.CreateStraightWireCable(....
```

And then, when a Harness has been created from various components, it is inserted to the Modeler:

```
oEditor = oDesign.SetActiveEditor("3D Modeler")

oEditor.InsertNativeComponent(

      [

            "NAME:InsertNativeComponentData",

            "TargetCS:=" , "Global",

            "SubmodelDefinitionName:=", "MyHarness",
```

...

AddCableToBundle

CreateCableBundle

CreateCableHarness

CreateClockSource

CreatePWLSource

CreateStraightWireCable

CreateTwistedPairCable

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportCableLibrary

ImportCableLibrary

RemoveCable

UpdateCableHarness

AddCableToBundle

Add a cable to an existing bundle using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables** > **Edit Cables** to open the **Cable Editor**.

UI Access	**3D Components** > **Cables** > **Edit Cables**.
Parameters	None
Return Value	None

Python Syntax	AddCableToBundle(<parameters>))

Python Example	`oModule.AddCableToBundle("bundle1", "tpwire2", 1,`
	`  [`
	`  "NAME:CableInstParams",`
	`  "XPos:=" , "0mm",`
	`  "YPos:=" , "0mm",`
	`  "RotX:=" , "0deg"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
                "NAME:CableInstAttribs",
                "Name:="                , "tpwire2"
        ])
```

CreateCableBundle

Creates a cable bundle using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>Edit Cables** to open the **Cable Editor**. Then **Add>Bundle...**.

UI Access	**3D Components>Cables** > **Edit Cables**
Parameters	None
Return Value	None

Python Syntax	CreateCableBundle([<parameters>)], [<name>])
Python Example	oModule.CreateCableBundle(["NAME:BundleParams", "AutoPack:=" , True, ["NAME:InsulationJacketParams", "InsThickness:=" , "0.25mm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "JacketMaterial:="      , "PVC plastic",
                        "InnerDiameter:="       , "2.5mm"
                ]
        ],
        [
                "NAME:BundleAttribs",
                "Name:="                , "bundle1"
        ])
```

CreateCableHarness

Creates a cable harness using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>New Cable Harness** to open the **Insert Cable Harness Component Editor**.

UI Access	**3D Components>Cables** > **New Cable Harness...**
Parameters	None
Return Value	None

Python Syntax	CreateCableHarness(<parameters>))
Python Example	`oModule.CreateCableHarness(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:HarnessParams",
                "Bundle:="              , "bundle1",
                "TwistAlongPath:="      , "720deg",
                "Route:="               , "Polyline1",
                "AutoOrient:="          , False,
                "Origin:="              , ["0mm","0mm","0mm"],
                "XAxisEnd:="            , ["-6mm","0mm","0mm"],
                "PlaneFlip:="           , True,
                [
                        "NAME:RefConductors",
                        "tpwire2/w1"
                ],
                [
                        "NAME:InputTerminations",
                        "tpwire2/w1:="          , [                     "Source:="
                        "tpwire2/w2:="          , [                     "Imped:="
                ],
                [
                        "NAME:OutputTerminations",
                        "tpwire2/w1:="          , [                     "Imped:="
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "tpwire2/w2:="          , [                             "Source:="
                ]
        ],
        [
                "NAME:HarnessAttribs",
                "Name:="                , "MyHarness"
        ])
```

CreateClockSource

Creates a clock source using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>Edit Cable Sources...** to open the **Cable Time Domain Sources Editor**.

UI Access	**3D Components>Cables** > **Edit Cables**
Parameters	None
Return Value	None

Python Syntax	CreateClockSource(<parameters>))
Python Example	`oModule.CreateClockSource(` `[`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NAME:ClockSignalParams",
                "Period:="              , "35us",
                "LowPulseVal:="         , "0V",
                "HighPulseVal:="        , "1V",
                "Risetime:="            , "5us",
                "Falltime:="            , "5us",
                "PulseWidth:="          , "20us"
        ],
        [
                "NAME:TDSourceAttribs",
                "Name:="                , "clock1"
        ])
```

CreatePWLSource

Creates a PWL source using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables** > **Edit Cable Sources...** to open the **Cable Time Domain Sources Editor**.

UI Access	**3D Components>Cables** > **Edit Cable Sources...**
Parameters	None
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreatePWLSource(<parameters>))
Python Example	oModule.CreatePWLSource(["NAME:PWLSignalParams", ["NAME:SignalValues", "0V", "0.5V", "0V"], ["NAME:TimeValues", "0ns", "1ns", "2ns"]], ["NAME:TDSourceAttribs", "Name:=" , "pwl3"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreateStraightWireCable

Creates a straight Wire Cable using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables** > **Edit Cables** to open the **Cable Editor**. Then **Add Straight Wire Cable**.

UI Access	**3D Components>Cables** > **Edit Cables**
Parameters	None
Return Value	None

Python Syntax	CreateStraightWireCable([<parameters>)], [<name>])
Python Example	oModule = oDesign.GetModule("CableSetup") oModule.CreateStraightWireCable(["NAME:StWireParams", "WireStandard:=" , "ISO", "WireGauge:=" , "0.13", "CondDiameter:=" , "0.55mm", "CondMaterial:=" , "copper", "InsThickness:=" , "0.25mm", "InsMaterial:=" , "PVC plastic",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"InsType:=" , "Thin Wall"], ["NAME:StWireAttribs", "Name:=" , "stwire1"])

CreateTwistedPairCable

Creates a twisted pair cable using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>Edit Cables** to open the **Cable Editor**. Then **Add Twisted Pair Cable**.

UI Access	**3D Components>Cables** > **Edit Cables**
Parameters	None
Return Value	None

Python Syntax	CreateTwisted Pair Cable([<parameters>)], [<name>])
Python Example	oModule.CreateTwistedPairCable("stwire1", ["NAME:TwistedPairParams",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:VirtualJacketParams",
                "JacketMaterial:="      , "",
                "InnerDiameter:="       , "0"
        ],
                "IsLayLengthSpecified:=", False,
                "LayLength:="           , "13.8888888888889mm",
                "TurnsPerMeter:="       , "72"
        ],
        [
                "NAME:TwistedPairAttribs",
                "Name:="                , "tpwire1"
        ])
```

ExportCableLibrary

Exports a cable library created using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables** > **Export Cable Library.** to open a browser window.

UI Access	**3D Components** > **Cables** > **Export Cable Library...**
Parameters	None
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportCableLibrary(<parameters>))
Python Example	`oModule = oDesign.GetModule("CableSetup")` `oModule.ExportCableLibrary("D:\\Users\\JaneDoe\\PersonalLib\\MyCables")`

ImportCableLibrary

Imports an existing cable library created using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>Export Cable Library.** to open a browser window.

UI Access	**3D Components>Cables** > **Import Cable Library...**
Parameters	None
Return Value	None

Python Syntax	ImportCableLibrary(<parameters>))
Python Example	`oModule = oDesign.GetModule("CableSetup")` `oModule.ExportCableLibrary("D:\\Users\\JaneDoe\\PersonalLib\\MyCables")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveCable

Removes an existing cable created using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>Edit Cables....** to open a **Cable Editor** window.

UI Access	**3D Components>Cables** > **Edit Cables...**
Parameters	None
Return Value	None

Python Syntax	RemoveCable(<parameters>))
Python Example	`oModule = oDesign.GetModule("CableSetup")` `oModule.RemoveCable("bundle4")`

UpdateCableHarness

Creates a cable harness using the Beta Cable Modeling feature. Right click on the 3D Component icon and select **Cables>New Cable Harness** to open the **Insert Cable Harness Component Editor**.

UI Access	**3D Components>Cables** > **New Cable Harness...**
Parameters	None
Return Value	None

Python Syntax	UpdateCableHarness(<parameters>))

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example

```
oModule.UpdateCableHarness(
        [
                "NAME:HarnessParams",
                "Bundle:="              , "bundle1",
                "TwistAlongPath:="      , "720deg",
                "Route:="               , "Polyline1",
                "AutoOrient:="          , False,
                "Origin:="              , ["0mm","0mm","0mm"],
                "XAxisEnd:="            , ["-6mm","0mm","0mm"],
                "PlaneFlip:="           , True,
                [
                        "NAME:RefConductors",
                        "tpwire2/w1"
                ],
                [
                        "NAME:InputTerminations",
                        "tpwire2/w1:="          , [                             "Source:="
                        "tpwire2/w2:="          , [                             "Imped:="
                ],
                [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "NAME:OutputTerminations",
                        "tpwire2/w1:="          , [                         "Imped:="
                        "tpwire2/w2:="          , [                         "Source:="
                ]
        ],
        [
                "NAME:HarnessAttribs",
                "Name:="                , "MyHarness"
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

9 - Output Variable Script Commands

Output variable commands should be executed by the "OutputVariable" module.

First, obtain the output variable module from oDesign and use it for output variable commands:

```
oModule = oDesign.GetModule("OutputVariable")
oModule.<CommandName><args>
```

The commands are:

DeleteOutputVariable

DoesOutputVariableExist

EditOutputVariable

ExportOutputVariables

GetOutputVariableValue

GetOutputVariables

ImportOutputVariables

CreateOutputVariable

Adds a new output variable. Different forms of this command are executed for different design types.

> **Note:**
>
> Output variables are associated with a name and an expression. The name is not permitted to collide with design variable, sim value, or other output variable names. It cannot have spaces or any arithmetic or other operators in it. Expression definitions cannot be cyclic. For example, A = 2*B, B=3*A is not allowed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>HFSS 3D Layout > Results > Output Variables.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><OutputVarName></td><td>String</td><td>Name of the new output variable.</td></tr>
<tr><td><Expression></td><td>Value</td><td>Value to assign to the variable.</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution, as seen in the output variable UI.</td></tr>
<tr><td><reportType> or <solutionType></td><td>String</td><td>The name of the report type as seen in the output variable UI.

For Layout Editor, use the solution type.</td></tr>
<tr><td><ContextArray> or <DomainArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated.

<code>["Context:=", <string>]</code>

For Layout Editor, use Domain array:

<code>["Domain:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>CreateOutputVariable (<i><OutputVarName></i>, <i><Expression></i>, <i><SolutionName></i>, <i><reportType</i> | <i>solutionType></i>, <i><contextArray</i> | <i>domainArray></i>)</td></tr>
<tr><td>Python Example</td><td>HFSS Example:

<code>oModule.CreateOutputVariable("Var" & OutputQuantity, OutputQuantity, Solution, "Far Fields", ["Context:=", InfiniteSphere, "Domain:=", "Sweep"])</code></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Layout Editor Example :
	`oModule = oDesign.GetModule("OutputVariable")`
	`oModule.CreateOutputVariable("test","mag(S(WavePort1,WavePort1))", "Setup1 : LastAdaptive", "Modal Solution Data", ["Domain:=", "Sweep"])`

DeleteOutputVariable

Deletes an existing output variable. The variable can only be deleted if it is not in use by any traces.

UI Access	**HFSS 3D Layout** > **Results** > **Output Variables**. In the **Output Variables** window, click **Delete**.
Parameters	Name | Type | Description <*OutputVarName*> | String | Name of the output variable
Return Value	None.

Python Syntax	DeleteOutputVariable (<*OutputVarName*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.DeleteOutputVariable ("testNew")`

DoesOutputVariableExist

Verifies whether or not a named output variable exists.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*outputVariableName*></td><td>String</td><td>Output variable name</td></tr></table>
Return Value	`True` if the variable exists; `False` if it does not.

Python Syntax	DoesOutputVariableExist(<*outputVariableName*>)

Python Example	`oProject = oDesktop.GetActiveProject()`
	`oDesign = oProject.GetActiveDesign()`
	`oModule = oDesign.GetModule("OutputVariable")`
	`oModule.DoesOutputVariableExist("MyTestVar")`

EditOutputVariable

Changes the name or expression of an existing output variable.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*OrigVarName*></td><td>String</td><td>Name of the original output variable</td></tr><tr><td><*NewExpression*></td><td>String</td><td>New value to assign to the variable</td></tr><tr><td><*NewVarName*></td><td>String</td><td>New name of the variable if any, else pass empty string</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of the solution as seen in the output variable UI. For example, "`Setup1 : Last Adaptive`".</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*ReportType*> or <*SolutionType*>	String	Name of the report type as seen in the output variable UI For Layout Editor, use the solution type.

	<*ContextArray*> or <*DomainArray*>	Array	Structured array containing context for which the output variable expression is being evaluated.
			
			`["Context:=", <string>]`
			
			For Layout Editor, use Domain array:
			
			`["Domain:=", <string>]`

Return Value	None		

Python Syntax	EditOutputVariable (<*OrigVarName*>, <*NewExpression*>, <*NewVarName*>, <*SolutionName*>, <*ReportType* | *SolutionType*>, <*ContextArray* | *DomainArray*>)

Python Example	`oModule = oDesign.GetModule("OutputVariable")`
	
	`oModule.EditOutputVariable ("test", "normalize(R1_0.V)", "testNew", "TR", "Standard", [])`

ExportOutputVariables

Exports output variables to a file.

UI Access	Click on **Export** in the **Output Variables** dialog.		
Parameters	Name	Type	Description
	<*FileName*>	String	Name of the file include path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Boolean: • **True** - output variable successfully exported. • **False** - error when export output variables.

Python Syntax	ExportOutputVariables(<*FileName*>)
Python Example	`oModule.ExportOutputVariables("C:/output_var.aoutvar")`

GetOutputVariables

Returns the list of output variables.

UI Access	N/A
Parameters	None.
Return Value	Array containing all output variables.

Python Syntax	GetOutputVariables()
Python Example	`oDesign.GetOutputVariables()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetOutputVariableValue

Returns the double value of an output variable. Only expressions that return a double value are supported. The expression is evaluated only for a single point.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OutputVarName></td><td>String</td><td>Name of the output variable.</td></tr>
<tr><td><IntrinsicVariation></td><td>String</td><td>A set of intrinsic variable value pairs to use when evaluating the output expression.
HFSS example:
<code>"Freq='20GHz' Theta='20deg' Phi='30deg'"</code></td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution as listed in the output variable UI. For example, <code>"Setup1 : Last Adaptive"</code>.</td></tr>
<tr><td><ReportType></td><td>String</td><td>The name of the report type as seen in the output variable UI.
Possible values in Layout/Schematic Editor are:
"Standard"
"Load Pull"
"Constellation"
"Data Table"
"Eye Diagram"
"Statistical"</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated. Can be empty.
<code>["Context:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Double value of the output variable.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetOutputVariableValue(<*OutputVarName*>, <*IntrinsicVariation*>, <*SolutionName*>, <*ReportTypeName*>, <*ContextArray*>)
Python Example	`Val = oDesign.GetOutputVariableValue("test","Freq = '20Ghz' Theta='20deg' Phi='30deg'", "TR", "Standard", [])`

ImportOutputVariables

Imports output variables from a file.

UI Access	Click on **Import** in the **Output Variables** dialog.
Parameters	Name | Type | Description <*FileName*> | String | Name of the file include path.
Return Value	Boolean: • **True** - output variable successfully imported. • **False** - error when import output variables.

Python Syntax	ImportOutputVariables(<*FileName*>)
Python Example	`oModule.ImportOutputVariables("C:/output_var.aoutvar")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SimValueContext

SimValueContext holds context information for a trace, and describes how data for a trace should be extracted from the simulation. SimValueContext contains a list of 14 required initial values:

```
SimValueContext (
Domain ID, Calculation Type, Number of Cycles, Rise Time,
Step, Impulse, Context ID, Window Width,
Window Type, TDR Kaiser Parameter, Hold Time, DeviceName,
TDR Step Time, DR Maximum Time )
```

For example, the following indicates a trace in the Time Domain, Standard Calculation with the number of cycles being 2:

```
"SimValueContext:=", Array(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0)
```

Additional, context-specific values may follow the required values, as described in subsection 15 below.

1. Domain ID

No Domain	0
Time Domain	1
Spectrum Domain	2
Sweep Domain	3
Device Domain	4
SinglePt Domain	5
LoadPull Domain	6
Transient Domain	7
Budget Domain	8

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

NetworkFunction Domain	9
Oscillator Domain	55802
Noise Domain	55803
Transfer Function Domain	55807
Time Frequency Domain	55808
Transient Time Domain	55809
Periodic AC Domain	55818
UI Domain	55819
Eye Measurement Domain	55823
Initial Response Domain	55824
Peak Distortion Domain	55825

2. Calculation Type

Standard Calculation	0
Device2_DCIV	1
Device3_DCIV_Output	2
Device3_DCIV_Input	3
Device3_DCIV_Transfer	4
Device3_DCIV_Reverse	5
Device2_ACLoad	6
Device3_ACLoad_Output	7
Device3_ACLoad_Input	8

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Device3_ACLoad_Transfer	9
Device3_ACLoad_Reverse	10
Constellation	11
EyeDiagram	12
FreeX (Statistic Report)	13

3. **Number of Cycles** — Used in Time Domain in HarmonicBalance analysis.
4. **Rise Time** — Not used by Designer/Nexxim.
5. **Step** — Not used by Designer/Nexxim.
6. **Impulse** — Not used by Designer/Nexxim.
7. **Context ID** — Not used by Designer/Nexxim.
8. **Window Width** — Not used by Designer/Nexxim.
9. **Window Type** — Not used by Designer/Nexxim.
10. **TDR Kaiser Parameter** — Not used by Designer/Nexxim.
11. **Hold Time** — Not used by Designer/Nexxim.
12. **DeviceName** — Not used by Designer/Nexxim.
13. **TDR Step Time** — Not used by Designer/Nexxim.
14. **TDR Maximum Time** — Not used by Designer/Nexxim.
15. **Context-specific values** — Used in Time Domain in HarmonicBalance analysis.

Context-specific values are entered in the format "key, true/false, keyvalue", where:

— **"key"** is the name of the key being set.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

— **"true/false"** indicates whether the key is a string value.

— **"keyvalue"** is the value of the key.

— The order of the context keys is not significant.

— Context keys have software defaults that will be used if not provided in the script.

a. Plotting Range for Time domain in Transient and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Start Time	WS	False	0ns
Stop Time	WE	False	10ns
Minimum Time	WM	False	0ns
Maximum Time	WN	False	10ns
Is Thinning Enabled?	DE	False	0
Dy/dx Tolerance	DT	False	0.001
Number of points	DP	False	20000000

b.Transient report context for Spectral domain in Transient analysis:

Description	Key Name	Is a string?	Key Value
Start Time	TS	False	0ns

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Stop Time	TE	False	10ns
Max Harmonics	MH	False	100
Max Frequency	MF	False	*
Window type	WT	False	0
Width Percentage	WW	False	100
Kaiser Parameter	KP	False	0
Adjust Coherent Gain	CG	False	0

* Script can specify either MH or MF. If neither is specified, Max Harmonics is set to 100. If both are specified, MF is used.

Window Type	**ID**
Rectangular	0
Bartlett	1
Blackman	2
Hamming	3
Hanning	4
Kaiser	5
Welch	6
Weber	7
Lanzcos	8

c. Eyeprobe index context for UI domain, Time domain, Eye Measuremant domain in VerifEye and QuickEye analysis:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Description	Key Name	Is a string?	Key Value
Eyeprobe compinst ID	PCID	False	0

d. Eyesource index context for Initial Response domain and Peak Distortion domain in VerifEye and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Eyesource compinst ID	SCID	False	0

e. UI domain context in VerifEye and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Use midpoint?	MIDPOINT	False	0 - Don't use midpoint. 1 - Use midpoint of amplitude. 2 - Use midpoint of UI.
Minimum latch overdrive	MLO	False	0

f. Distribution Context for UI Domain in VerifEye and QuickEye analysis:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Description	Key Name	Is a string?	Key Value
Use distribution?	USE_DIST	False	0 - No 1 - Yes
Distribution type	DIST	False	0 - Receiver Jitter 1 - Receiver Noise 2 - User Defined

Receiver Jitter Parameters

Description	Key Name	Is a string?	Key Value
DLL standard deviation	DSD	False	0
Distribution type	DIST	False	0
DLL taps	DMN	False	0
Static Offset	SOFF	False	0
Number of Gaussian data sets	NUMG	False	0
Gaussian std deviation	GS0,GS1...	False	0
Offset mean	GM1,GM1...	False	0
Number of Uniform data sets	NUMU	False	0
Uniform width	UW0,UW1...	False	0
Uniform mean	UM1,UM1...	False	0

Receiver Noise Parameters

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Description	Key Name	Is a string?	Key Value
Number of Gaussian data sets	NUMG	False	0
Gaussian std deviation	GS0,GS1...	False	0
Number of Uniform data sets	NUMU	False	0
Uniform width	UW0,UW1...	False	0

User Defined Parameters

Description	Key Name	Is a string?	Key Value
Number of XY data pairs	NUMG	False	0
X data	X0,X1,X2...	False	0
Y data	Y0,Y1,Y2...	False	0
Cutoff probability	CP	False	0

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

10 - Reporter Editor Script Commands

Reporter commands should be executed by the oDesign object.

For example:

```
oDesign = Project.SetActiveDesign("EMDesign1")

oModule = oDesign.GetModule("ReportSetup")
```

All Report and Trace properties can be edited using the **ChangeProperty** commands. This includes Title properties, General properties, and Background properties such as border color, fonts, X and Y axis scaling, and number display.

> **Note:**
>
> When you execute **Tools** > **Record Script**, operations performed in the Reporter are automatically recorded.

The list of commands is as follows:

AddAllEyeMeasurements

AddCartesianLimitLine

AddCartesianLimitLineFromCurve

AddCartesianLimitLineFromEquation

AddCartesianXMarker

AddCartesianYMarker

AddCartesianYMarkerToStack

AddDeltaMarker

AddMarker

AddNote

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddTraceCharacteristics

AddTraces

AddVerifEyeAnalysis

ApplyReportTemplate

ChangeProperty

ClearAllMarkers

ClearAllTraceCharacteristics

CloneReportsFromDatasetSolution

CopyPlotSettings

CopyReportDefinitions

CopyReportData

CopyTraceDefinitions

CopyTracesData

CreateReportFromFile

CreateReportFromTemplate

CreateReportOfAllQuantities

DeleteAllReports

DeleteReports

DeleteTraceCharacteristics

DeleteTraces

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DoesSupportTraceCharacteristics

DumpAllReportsData

EditCartesianXMarker

EditCartesianYMarker

EditMarker

EditQuickEyeAnalysis

EditVerifEyeAnalysis

ExportEyeMaskViolation

ExportImageToFile

ExportReportDataToFile

ExportTableToFile

ExportToFile [Reporter]

GetAllCategories

GetAllQuantities

GetAllReportNames

GetAvailableDisplayTypes

GetAvailableReportTypes

GetAvailableSolutions

GetChildNames

GetChildObject

GetChildTypes

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetCurvePropServerName

GetDisplayType

GetDynLinkIntrinsicVariables

GetDynLinkQtyValueState

GetDynLinkLinkTraces

GetDynLinkVariableValues

GetName

GetObjPath

GetPropNames

GetPropSIValue

GetPropertyValue

GetQtyExpressionsForSourceTrace

GetReportSummaryForRegressionTesting

GetReportTraceNames

GetSolutionContexts

GetSolutionDataPerVariation

GroupPlotCurvesByGroupingStrategy

ImportIntoReport

ImportReportDataIntoReport

MovePlotCurvesToGroup

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

MovePlotCurvesToNewGroup

OpenWindowForAllReports

OpenWindowForReports

PastePlotSettings

PasteReports

PasteReportsWithLegacyNames

PasteTraces

PasteTracesWithLegacyNames

RenameReport

RenameTrace

ResetPlotSettings

SavePlotSettingsAsDefault

SetLinkOutputTraces

SetPropValue

UngroupPlotCurvesInGroup

UpdateAllFieldsPlots

UpdateAllReports

UpdateReports

UpdateTraces

UpdateTracesContextandSweeps

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddAllEyeMeasurements

Displays all the eye measurements in tabular format.

UI Access	Right-click the report and select **Trace Characteristics** > **Add All Eye Measurements**
Parameters	Name | Type | Description *<ReportName>* | String | Name of the report.
Return Value	None.

Python Syntax	AddAllEyeMeasurements(*<ReportName>*)
Python Example	`oModule.AddAllEyeMeasurements("DQS Eye")`

AddCartesianLimitLine

Adds a limit line to a report on the X axis.

UI Access	**Report2D** > **Add Limit Line**> **Specify Points...**

Parameters	Name | Type | Description
	<ReportName> | String | Name of the report.
	<Def> | Array | Structured array:
	`["NAME:CartesianLimitLine",`
	`["NAME:XValues", <integer X values>],`
	`"XUnits:=", <string unit of measure for X>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:YValues", <integer Y values>], "YUnits:=", "<string unit of measure for Y>", "YAxis:=", <string name of associated Y axis>]
Return Value	None.

Python Syntax	AddCartesianLimitLine (<*ReportName*>, <*Def*>)
Python Example	oModule.AddCartesianLimitLine("Project Outputs", ["NAME:CartesianLimitLine", ["NAME:XValues",0, 2, 5, 7, 10, 15], "XUnits:=", "s", ["NAME:YValues",0.05, 0.3, 0.65, 0.825, 0.95, 1], "YUnits:=", "mV", "YAxis:=", "Y1"])

AddCartesianLimitLineFromCurve

Adds a limit line to a report from selected curve on the plot.

UI Access	**Report2D** > **Add Limit Line** > **From Selected Curve...**
Parameters	Name / Type / Description <*ReportName*> / String / Name of the report. <*LimitLineParams*> / String / Structured array. ["NAME:CartesianLimitLineFromCurve",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"TraceName:=", <string name of selected trace>, "CurveName:=", <string name of selected curve>, "Start:=", "<value><unit>", "Stop:=", "<value><unit>", "YAxis:=", <integer Y-Axis number>, "YOffset:=", <value offset from Y-Axis>, "CreateMode:=", "<AboveCurve | BelowCurve | Above and Below Curve>", "YShiftPercent:=", <value percentage to shift from Y>]
Return Value	None.		

Python Syntax	AddCartesianLimitLineFromCurve(<*ReportName*>, <*LimitLineParams*>)
Python Example	oModule.AddCartesianLimitLineFromCurve("Variables Plot 2", ["NAME:CartesianLimitLineFromCurve", "TraceName:=" , "Phase", "CurveName:=" , "", "Start:=" , "0deg", "Stop:=" , "375deg", "YAxis:=" , 1,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"YOffset:=" , 0, "CreateMode:=" , "AboveCurve", "YShiftPercent:=" , 10])

AddCartesianLimitLineFromEquation

Adds a limit line to a report from a specified equation.

UI Access	**Report2D** > **Add Limit Line** > **Specify Equation...**		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of the report.
	<*LimitLineParams*>	Array	Structured array. ["NAME:CartesianLimitLineFromEquation", "YAxis:=", <integer Y-Axis number>, "Start:=", <string start frequency with unit>, "Stop:=", <string end frequency with unit>, "Step:=", <string frequency resolution with unit>, "Equation:=", <string specified equation>) "XValuesUnit:=", "GHz"]
Return Value	None.		

Python Syntax	AddCartesianLimitLineFromEquation(<*ReportName*>, <*LimitLineParams*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oModule.AddCartesianLimitLineFromEquation("S Parameter Plot 1", ["NAME:CartesianLimitLineFromEquation", "YAxis:=" , 1, "Start:=" , "9GHz", "Stop:=" , "11GHz", "Step:=" , "0.2GHz", "Equation:=" , "x+1" "XValuesUnit:=" ,"GHz"])

AddCartesianXMarker

Adds a marker to a report on the X axis.

UI Access	**Report2D** > **Marker** > **Add X Marker**.		
Parameters	Name	Type	Description
	<ReportName>	String	Name of report.
	<MarkerName>	String	Marker name, including any trailing number.
	<XValue>	Double	X coordinate.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddCartesianXMarker (<*ReportName*>, <*MarkerName*>, <*XValue*>)
Python Example	`oModule.AddCartesianXMarker ("XY Plot 1", "MX1", 0)`

AddCartesianYMarker

Adds a marker to a report on the Y axis.

UI Access	**Report2D** > **Marker** > **Add Y Marker**.
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report.
<*MarkerName*>	String	Marker name, including any trailing number.
<*AxisName*>	String	Name of axis.
<*YValue*>	Double	Y coordinate.
<*CurveName*>	String	Name of curve.

Return Value	None

Python Syntax	AddCartesianYMarker (<*ReportName*>, <*MarkerName*>, <*AxisName*>, <*YValue*>, <*CurveName*>)
Python Example	`oModule.AddCartesianYMarker("XY Plot 1", "MY1", "Y1", 0,` `"dB() : Setup1 : Sweep1")`

AddCartesianYMarkerToStack

Adds a marker to a stacked report on the Y axis.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report.</td></tr><tr><td><*MarkerName*></td><td>String</td><td>Marker name, including any trailing number.</td></tr><tr><td><*AxisName*></td><td>String</td><td>Name of axis.</td></tr><tr><td><*YValue*></td><td>Double</td><td>Y coordinate.</td></tr><tr><td><*CurveName*></td><td>String</td><td>Name of curve.</td></tr><tr><td><*StackOption*></td><td>Array</td><td>"Current" to create marker on the current stack. "All" to create marker on all stack.</td></tr></table>
Return Value	None

Python Syntax	AddCartesianYMarkerToStack (<*ReportName*>, <*MarkerName*>, <*AxisName*>, <*YValue*>, <*CurveName*>, <*StackOption*>)
Python Example	`oModule.AddCartesianYMarker("XY Plot 1", "MY1", "Y1", 0,` `"dB() : Setup1 : Sweep1", ["All"])`

AddDeltaMarker

Add markers to calculate differences between two trace points on a plot.

UI Access	**Report2D** > **Marker** > **Add Delta Marker**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report.</td></tr><tr><td><*MarkerName1*></td><td>String</td><td>Marker name, including any trailing number, for the first marker.</td></tr><tr><td><*CurveName1*></td><td>String</td><td>Full trace name for the first marker.</td></tr><tr><td><*PrimarySweepValue1*></td><td>String</td><td>Frequency, including unit, for the first marker's sweep.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*MarkerName2*>	String	Marker name, including any trailing number, for the second marker.
	<*CurveName2*>	String	Full trace name for the second marker.
	<*PrimarySweepValue2*>	String	Frequency, including unit, for the second marker's sweep.
Return Value	None		

Python Syntax	AddDeltaMarker(<*ReportName*>, <*MarkerName1*>, <*CurveName1*>, <*PrimarySweepValue1*>, <*MarkerName2*>, <*CurveName2*>, <*PrimarySweepValue2*>)

Python Example	`oModule.AddDeltaMarker("S Parameter Plot 1", "m1",`
	`"dB(S(A-MII-RXD1_30_SQFP28X28_208_U1 A-MII-RXD1_30_SQFP28X28_208_U1)) : SYZ Sweep 1 : SYZ Sweep 1 : Cartesian",`
	`"1.7GHz", "m2",`
	`"dB(S(A-MII-RXD1_30_SQFP28X28_208_U1 A-MII-RXD1_30_SQFP28X28_208_U1)) : SYZ Sweep 1 : SYZ Sweep 1 : Cartesian",`
	`"5.05GHz")`

AddMarker

Adds a marker to a trace on a report.

UI Access	**Report2D** > **Marker** > **Add Marker**.		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of report.
	<*MarkerName*>	String	Marker name, including any trailing number.
	<*CurveName*>	String	Full trace name.
	<*PrimarySweepValue*>	String	Primary sweep value, including unit.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	AddMarker(<*ReportName*>, <*MarkerName*>, <*CurveName*>, <*PrimarySweepValue*>)
Python Example	`oModule.AddMarker("XY Plot 1", "m5",` `"GS1.VAL : TR4 : Cartesian", "3.61599999999997s")`

AddNote

Adds a note at a specified location to a given report.

UI Access	Right-click on the plot and select **Add Note**.

Parameters

Name	Type	Description
<*ReportName*>	String	Name of report
<*NoteDataArray*>	Array	Structured array. `["NAME:<StringDataName>", <NoteArray>]`

<*NoteArray*>	Array	Structured array:
		`["NAME:<NoteDataSourceName>",`
		`"SourceName:=", <string source name>,`
		`"HaveDefaultPos:=", <boolean True for position 0,0; False to specify below>,`
		`"DefaultXPos:=", <int X position for note; 0 for default>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"DefaultYPos:=", <int Y position for note; 0 for default>, "String:=", <string note text>]
Return Value	None		

Python Syntax	AddNote (<*ReportName*>, <*NoteDataArray*>)
Python Example	oModule.AddNote("XY Plot1", ["NAME:NoteDataSource", "SourceName:=", "Note1", "HaveDefaultPos:=", False, "DefaultXPos:=", 1996, "DefaultYPos:=", 3177, "String:=", "This is a note."])

AddTraceCharacteristics

Adds a trace characteristics field to the legend on a report.

UI Access	**Report2D** > **Trace Characteristics** > **All**. This opens the **Add Trace Characteristics** dialog box.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*ReportName*>	String	Name of report.
<*FunctionName*>	String	The function name. See the **Functions** column of the **Add Trace Characteristics** dialog box.

<*FunctionArgs*>	Array	Array containing string values for any function arguments. Pass empty array if no arguments exist.
		To see which argument(s) a function takes, see the bottom of the **Add Trace Characteristics** dialog box.
		For function with one argument:
		`[<value>]`
		For function with multiple arguments:
		`[<value>, <value>, ...]`

<*RangeArgs*>	Array	Required. Array containing either string "Full" for a full sweep range, or "Specified" and strings containing the start and end values for the frequency range. For example: `["Specified", "19.5GHz", "24.4GHz"]`

Return Value

None.

Python Syntax	AddTraceCharacteristics (<*ReportName*>, <*FunctionName*>, <*FunctionArgs*>, <*RangeArgs*>)
Python Example	`oModule.AddTraceCharacteristics("XY Plot 2", "delaytime", ["0"], ["Full"])` `oModule.AddTraceCharacteristics("Differential S-parameters", "prms", ["0", "0"], ["Full"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oModule.AddTraceCharacteristics("Rept2DRectFreq", "distortion", ["0"], ["Specified", "19.5GHz", "20.4GHz"])`

AddTraces

Creates a new trace and adds it to the specified report.

UI Access	**Modify Report** > **Add Trace**.
Parameters	See table below

Name	Type	Description
<*ReportName*>	String	Name of Report
<*SolutionName*>	String	Name of the solution as listed in the **Modify Report** dialog box.

<*ContextArray*>	Array	Context for which the expression is being evaluated. This can be an empty string if there is no context.
		`["Domain:=", <DomainType>]`
		`<DomainType> ex. "Sweep" or "Time"`
		`["Context:=", <GeometryType>]`
		`<GeometryType> ex. "Infinite Sphere`*n*`", "Sphere`*n*`", "Polyline`*n*`"`

<*FamiliesArray*>	Array	Contains sweep definitions for the report.
		`["<VariableName>:= ", <ValueArray>]`
		`<ValueArray>`
		`["All"] or ["Value1", "Value2", ..."Value`*n*`"]`
		`examples of <VariableName> "Freq", "Theta", "Distance"`

<*ReportDataArray*>	Array	This array contains the report quantity and X, Y, and (Z) axis definitions.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["X Component:=", <VariableName>, "Y Component:=", <VariableName> | <ReportQuantityArray>] <ReportQuantityArray> ex. ["dB[S[Port1, Port1]]"]
Return Value	None		

Python Syntax	Add Traces(<*ReportName*>, <*SolutionName*>, <*ContextArray* >, <*FamiliesArray*>, <*ReportDataArray*>)
Python Example	oModule.AddTraces("XY Plot1", "Setup1 : Sweep1", ["Domain:=", "Time", "HoldTime:=", 1, "RiseTime:=", 0, "StepTime:=", 6.24999373748E-012, "Step:=", False, "WindowWidth:=", 1, "WindowType:=", 0, "KaiserParameter:=", 1, "MaximumTime:=", 6.2437437437444E-009], ["Time:=", ["All"], "OverridingValues:=", ["0s", "6.24999373748188e-012s", ...]], ["X Component:=", "Time", "Y Component:=", ["TDRZ(WavePort1)"]], [])

ApplyReportTemplate

Applies settings to a report from a template file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click on a report, select **Report Templates** > **Apply Settings**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of the report to apply settings to.</td></tr><tr><td><*TemplateFile*></td><td>String</td><td>Template file name with path.</td></tr><tr><td><*PropertyType*></td><td>String</td><td>Property types to apply. `("Graphical" | "Data" | "All")`</td></tr></table>
Return Value	None.

Python Syntax	ApplyReportTemplate(<*ReportName*>, <*TemplateFile*>, <*PropertyType*>)
Python Example	`oModule.ApplyReportTemplate("Variables Plot 1", "C:/MyTemplate.rpt", "Graphical")`

ChangeProperty[ReportSetup]

Change the properties for a Report.

UI Access	Double-click on a report to change its properties.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*PropertyArray*></td><td>Array</td><td>Varies, depending on the properties associated with the select object command.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ChangeProperty(<*PropertyArray*>)

Python Example	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(
	  [
	  "NAME:AllTabs",
	  [
	  "NAME:General",
	  [
	  "NAME:PropServers",
	  "XY Plot 1:General"
	  ],
	  [
	  "NAME:ChangedProps",
	  [
	  "NAME:Use Scientific Notation",
	  "Value:=", True
	  ]
	  ]
	  ]
	])
	
	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(
	  [
	  "NAME:AllTabs",
	  [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:Legend",
        [
        "NAME:PropServers",
        "S Parameter Plot 1:Legend"
        ],
        [
        "NAME:ChangedProps",
        [
        "NAME:Legend Name",
        "Value:=" , "Ansys"
        ]
      ]
   ]
])
     oModule.ChangeProperty(
     [
       "NAME:AllTabs",
       [
         "NAME:General",
         [
           "NAME:PropServers",
           "S Parameter Plot 1:General"
         ],
     [
       "NAME:ChangedProps",
       [
         "NAME:Auto Scale Fonts",
         "Value:=" , False
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ]
      ]
    ]
])
```

Example For Rectangular Stacked Plot

```
oProject = oDesktop.SetActiveProject("antenna_test")
oDesign = oProject.SetActiveDesign("Design1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
   [
     "NAME:AllTabs",
     [
       "NAME:Scaling",
       [
         "NAME:PropServers",
         "S Parameter Plot 1:AxisY2:Stack1",
         "S Parameter Plot 1:AxisY2:Stack2"
       ],
       [
         "NAME:ChangedProps",
         [
           "NAME:Min",
           "Value:="               , "-60"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ]
      ]
    ]
  ])
```

Example Change eye rect and eye plot:

```
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("test_nport_synth2")
oDesign = oProject.SetActiveDesign("Circuit1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Grid",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:Grid:Eye"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Major grid line style",
          "Value:=" , "Dot"
        ],
        [
          "NAME:Minor grid line style",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "Value:=" , "Dot"
          ]
        ]
      ],
      [
        "NAME:Axis",
        [
          "NAME:PropServers",
          "QuickEye Voltage Plot 1:AxisX:Eye"
        ],
        [
          "NAME:ChangedProps",
          [
            "NAME:Axis Color",
            "R:=" , 255,
            "G:=" , 128,
            "B:=" , 128
          ]
        ]
      ],
      [
        "NAME:Axis",
        [
          "NAME:PropServers",
          "QuickEye Voltage Plot 1:AxisY1:Rect"
        ],
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "NAME:ChangedProps",
          [
            "NAME:Axis Color",
            "R:=" , 255,
            "G:=" , 128,
            "B:=" , 128
        ]
      ]
    ]
])
oModule.ChangeProperty(
   [
     "NAME:AllTabs",
     [
       "NAME:Grid",
       [
         "NAME:PropServers",
         "QuickEye Voltage Plot 1:Grid:Rect"
       ],
       [
         "NAME:ChangedProps",
         [
           "NAME:Major grid line style",
           "Value:=" , "LongDash"
         ],
         [
           "NAME:Minor grid line style",
           "Value:=" , "LongDash"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``
	            ]
	          ]
	       ]
	    ])
	``

ClearAllMarkers

Clears all markers from a report.

UI Access	**Report2D** > **Markers** > **Clear All**.
Parameters	Name / Type / Description <*ReportName*> / String / Name of Report
Return Value	None

Python Syntax	ClearAllMarkers(<*ReportName*>)
Python Example	`oModule.ClearAllMarkers("XY Plot 1")`

ClearAllTraceCharacteristics

Clears all trace characteristics from the legend in a report.

UI Access	**Report2D** > **Trace Characteristics** > **Clear All**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of the plot</td></tr></table>
Return Value	None

Python Syntax	ClearAllTraceCharacteristics(<*PlotName*>)
Python Example	`oModule.ClearAllTraceCharacteristics("XY Plot 1")`

CloneReportsFromDatasetSolution

Clones a report for a solveed solution from a dataset solution.

UI Access	Right-click the Project tree on the report and choose **Clone from Dataset Solution** > [Dataset_SolutionName]
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportsToClone*></td><td>Array</td><td>Array of report names to clone.</td></tr><tr><td><*SoluNameToUse*></td><td>String</td><td>Name of the dataset solution.</td></tr></table>
Return Value	None.

Python Syntax	CloneReportsFromDatasetSolution(<*ReportsToClone*>, <*SoluNameToUse*>)
Python Example	`oModule.CloneReportsFromDatasetSolution(["Rectangular Plot1"], "DatasetSolution_rsm_5812")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CopyPlotSettings

Copies settings of a specified plot.

UI Access	Right-click a report, select **Copy**
Parameters	Name | Type | Description <*ReportName*> | String | Name of specified report.
Return Value	None.

Python Syntax	CopyPlotSettings(<*ReportName*>)
Python Example	`oModule.CopyPlotSettings("Plot 1")`

CopyReportDefinitions

Copy the definition of a report for paste operations.

UI Access	Select a report in the Project tree, right-click and select **Copy Definition**
Parameters	Name | Type | Description <*ReportsArray*> | Array | Names of reports from which to copy the definitions
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CopyReportDefinitions(<*ReportsArray*>)
Python Example	`oModule.CopyReportDefinitions(["Transmission", "Reflection"])`

CopyReportsData

Copy all data corresponding to the specified reports.

UI Access	Select a report in the Project tree, right-click and select **Copy Data**
Parameters	

Name	Type	Description
<*ReportsArray*>	Array	Names of reports from which to copy data

Return Value	None

Python Syntax	CopyReportsData (<*ReportsArray*>)
Python Example	`oModule.CopyReportsData (["Transmission","Reflection"])`

CopyTraceDefinitions

Copy trace definitions for a paste operation.

UI Access	Select a trace in the Project tree, right-click and select **Copy Definition**
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of Report
<*TracesArray*>	Array	Trace definitions to copy

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	CopyTraceDefinitions(<*ReportName*>, <*TracesArray*>)
Python Example	`oModule.CopyTraceDefinitions ("Transmission",` `["mag(S(Port1,Port2))"])`

CopyTracesData

Copies trace data for a paste operation.

UI Access	Select a trace in the Project tree, right-click and select **Copy Data**.
Parameters	Name | Type | Description <*ReportName*> | String | Name of Report. <*TraceArray*> | Array | Trace definitions from which to copy corresponding data.
Return Value	None.

Python Syntax	CopyTracesData(<*ReportName*>, <*TracesArray*>)
Python Example	`oModule.CopyTracesData ("Transmission",` `["mag(S(Port1,Port2))"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateReport

Creates a new report with a single trace and adds it to the **Results** branch in the project tree.

<table>
<tr><td>UI Access</td><td>Right-click on Results > Create [Type] Report</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ReportName></td><td>String</td><td>Name of report.</td></tr>
<tr><td><ReportType></td><td>String</td><td>Type of report.
Possible values are:
"Modal S Parameters" - Only for Driven Modal solution-type problems with ports.
"Terminal S Parameters" - Only for Driven Terminal solution-type problems with ports.
"Eigenmode Parameters" - Only for Eigenmode solution-type problems.
"Fields"
"Far Fields" - Only for problems with radiation or PML boundaries.
"Near Fields" - Only for problems with radiation or PML boundaries.
"Emission Test"</td></tr>
<tr><td><DisplayType></td><td>String</td><td>Type of display.
If ReportType is "Modal S Parameters", "Terminal S Parameters", or "Eigenmode Parameters", then set to one of the following:
<code>"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Smith Chart", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot", "Rectangular Stacked Plot" or "3D Polar Plot".</code>
If <ReportType> is "Fields", then set to one of the following:</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "Rectangular Stacked Plot" or "3D Rectangular Bar Plot". If <ReportType> is "Far Fields" or "Near Fields", then set to one of the following: "Rectangular Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot" "Rectangular Stacked Plot" or "3D Polar Plot" If <ReportType> is "Emission Test", then set to one of the following: "Rectangular Plot" or "Data Table"
	<*SolutionName*>	String	Name of the solution as listed in the **Modify Report** dialog box.
	<*ContextArray*>	Array	Context for which the expression is being evaluated. This can be an empty string if there is no context. ["Domain:=", <DomainType>] <DomainType> ex. "Sweep" or "Time" ["Context:=", <GeometryType>] <GeometryType> ex. "Infinite Sphere*n*", "Sphere*n*", "Polyline*n*"
	<*FamiliesArray*>	Array	Contains sweep definitions for the report. ["<VariableName>:= ", <ValueArray>] <ValueArray> ["All"] or ["Value1", "Value2", ..."Value*n*"]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			examples of <VariableName> "Freq", "Theta", "Distance"
	<ReportDataArray>	Array	This array contains the report quantity and X, Y, and (Z) axis definitions. ["X Component:=", <VariableName>, "Y Component:=", <VariableName> | <ReportQuantityArray>] <ReportQuantityArray> ex. ["dB(S(Port1, Port1))"]
Return Value	None.		

Python Syntax	CreateReport(<*ReportName*>, <*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*ContextArray*>, <*FamiliesArray*>, <*ReportDataArray*>, <*ExtendedInfo*>)

Python Example	Several examples are given below: (nominal, Optimetrics, spectral, Partial Discharge, 3D rectangular Bar Plot with Change Bar Properties, and Add a new orientation to a 3D model and 3D report)
	Nominal Example:
	`oModule.CreateReport( "XY Plot 2", "Standard",`
	`"Rectangular Plot", "TR",`
	`[`
	`"NAME:Context",`
	`"SimValueContext:=", [], 1, 0, 2, 0, false,`
	`false, -1, 1, 0, 1, 1, "", 0, 0]`
	`],`
	`[`
	`"Time:=" , ["All"],`
	`"aaa:=" , ["Nominal"]`
	`],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "X Component:=", "Time",
                "Y Component:=", ["E1.I"]
        ],
        [])
```

Optimetrics Example:

```
oModule.CreateReport ("XY Plot 3", "Standard",
        "Rectangular Plot", "TR",
        [
                "NAME:Context",
                "OptiSetup:=", "ParametricSetup1", "SimValueContext:=",
                [
                        1, 0, 2, 0, false, false, 0, 1,0, 1, 1, "", 0, 0
                ]
        ],
        [
                "Time:=", ["All"], "aaa:=", ["Nominal"]
        ],
        [
                "X Component:="  , "Time",
                "Y Component:="  , ["E1.I"]
        ],
        [])
```

Spectral Example:

```
oModule.CreateReport ("XY Plot 4", "Standard",
        "Rectangular Plot", "TR",
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NAME:Context",
                "SimValueContext:=",
                [
                        2, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "",
                        0, 0, "CG", false, "0", "KP",false, "0", "MH",
                        false, "100", "TE", false, "40ms", "TH", false,
                        "40", "TS", false, "0ns", "UF", false, "0",
                        "WT", false, "0", "WW", false, "100"
                ]
        ],
        [
                "Spectrum:="   , ["All"], "aaa:=", ["Nominal"]
        ],
        [
                "X Component:=","Spectrum",
                "Y Component:=", ["mag(E1.I)"]
        ],
        [])
```

Partial Discharge Example

```
oModule.CreateReport("Partial Discharge Plot 2",
        "Partial Discharge", "Rectangular Plot",
        "PartialDischarge1 : PartialDischarge", [],
        [
                "GasPressure:=", ["All"],
                "Freq:="       , ["All"],
                "bend_angle:=" , ["All"]
        ],
        [
                "X Component:=", "GasPressure",
                "Y Component:=", ["Power"]
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example 3D Rectangular Bar Plot with Change Bar properties

```
oModule = oDesign.GetModule("ReportSetup")
oModule.CreateReport("S Parameter Plot 4", "Modal Solution Data",
       "3D Rectangular Bar Plot", "Setup1 : Sweep", [],
       [
              "Freq:="       , ["All"],
              "UU:="         , ["All"],
              "ZZZ:="        , ["Nominal"]
       ],
       [
              "X Component:=", "Freq",
              "Y Component:=", "UU",
              "Z Component:=", ["dB(S(wDipole1_1_p1,wDipole1_1_p1))"]
       ])

oModule.ChangeProperty(
       [
              "NAME:AllTabs",
              [
                     "NAME:Bar",
                     [
                            "NAME:PropServers"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Transparency",
                                        "Value:=", "0.5"
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Filled",
                                        "Value:=", False
                                ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                [
                                        "NAME:Show Outline",
                                        "Value:=", True
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Outline Color",
                                        "R:=", 0,
                                        "G:=", 0,
                                        "B:=", 160
                                ]
                        ]
                ]
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example Script to add a new orientation to a 3D model and 3D report

"AddViewOrientation" script command for 3D Report. The first argument must be the 3D report name.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oDesktop.OpenProject("D:/Tee.aedt")
oProject = oDesktop.SetActiveProject("Project")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.AddViewOrientation("Gain Plot 1", "3dreport", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.634171009063721,
      0.12737800180912,
      -0.762628972530365
    ],
    [
      "NAME:Up",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      -0.577874004840851,
      0.577228009700775,
      0.57694798707962
    ]
  ],
  [
    "NAME:ProjectionParams",
      [
      "NAME:Scaling",
      1.10227048397064,
      1.10227036476135,
      1.10227036476135
      ],
      [
      "NAME:Translation",
      -0.601842045783997,
      -0.352552056312561,
      -6.46677732467651
      ],
    "ProjectionType:=" , "Orthographic",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    [
      "NAME:ProjectionLimits",
      -2.53312301635742,
      2.53312301635742,
      -1,
      1,
      0.994183540344238,
      10.5401248931885
    ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.AddViewOrientation("3dmodel", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.627739012241364,
      -0.134194001555443,
      -0.766768991947174
    ],
    [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:Up",
      -0.694571018218994,
      0.54128497838974,
      0.473899990320206
    ]
  ],
  [
  "NAME:ProjectionParams",
    [
      "NAME:Scaling",
      0.0173665378242731,
      0.017366535961628,
      0.0173665322363377
    ],
    [
      "NAME:Translation",
      -0.0742612779140472,
      0.195189163088799,
      -5.89852476119995
    ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ProjectionType:=" , "Orthographic",
      [
        "NAME:ProjectionLimits",
        -2.27072739601135,
        2.27072739601135,
        -1,
        1,
        0.424737930297852,
        10.8078212738037
      ]
    ])
  oDesktop.CloseProject("Project")
```

CreateReportFromTemplate

Creates a report from a saved template.

UI Access	**[product]** > **Results** > **Report Templates** > **PersonalLib** > [TemplateName]
Parameters	Name | Type | Description *<TemplatePath>* | String | Path to report template
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateReportFromTemplate(<*TemplatePath*>)
Python Example	`oModule.CreateReportFromTemplate(` `"C:/MyHFSSProjects/PersonalLib/ReportTemplates/TestTemplate.rpt")`

CreateReportOfAllQuantities

Creates a report including all quantities in a category. Cannot create a report with expressions.

UI Access	NA
Parameters	

Name	Type	Description
<*ReportType*>	String	Report type name as input parameter
<*DisplayType*>	String	Display type name as input parameter
<*SolutionName*>	String	Solution name as input parameter
<*SimValueCtxt*>	String	A context name, or array of string that encoded the context(I).
<*CategoryName*>	String	Category name as input parameter
<*PointSet*>	Array	Array of strings(II)
<*CommonComponentsOfTraces*>	Array	Array of strings(III)
<*ExtTraceInfo*>	Array	Array of strings(IV)

Return Value	None.

Python Syntax	CreateReportOfAllQuantities(<*ReportNameArg*>, <*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*SimValueCtxt*>, <*CategoryName*>, <*PointSet*>, <*CommonComponentsOfTraces*>, <*ExtTraceInfo*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.CreateReportOfAllQuantities("Smith Chart all", "Modal Solution Data", "Smith Chart", "Setup1 : LastAdaptive", [],"S Parameter",["Freq:=", ["All"], "offset:=", ["All"],"a:=", ["Nominal"],"b:=", ["Nominal"]],[],[])`

DeleteMarker

Deletes the specified marker.

UI Access	[product] > **Fields** > **Fields** > **Marker** > **Delete Marker**.
Parameters	Name | Type | Description *<MarkerName>* | String | Name of the marker.
Return Value	None.

Python Syntax	DeleteMarker(*<MarkerName>*)
Python Example	`oModule.DeleteMarker("m1")`

DeleteAllReports

Deletes all existing reports.

UI Access	Right-click the report to delete in the project tree, and then click **Delete All Reports** on the shortcut menu.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	None.

Python Syntax	DeleteAllReports()
Python Example	`oModule.DeleteAllReports()`

DeleteReports

Deletes an existing report or reports.

UI Access	Right-click the report to delete in the project tree, and then click **Delete** on the shortcut menu
Parameters	Name | Type | Description *<ReportNameArray>* | Array | Array of report names to be deleted
Return Value	None.

Python Syntax	DeleteReports(<*ReportNameArray*>)
Python Example	`oModule.DeleteReports (["Rept2DRectFreq"])`

DeleteTraces

Deletes an existing traces or traces.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click the Trace to delete in the project tree, and then click **Delete** on the shortcut menu
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*TraceSelection*></td><td>Array</td><td>Structured array define selections. `["<ReportName>:=", <TracesArray>, <TracesArray>,... ]`</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of Report</td></tr><tr><td><*TracesArray*></td><td>Array</td><td>Contains the traces to delete within a report `[<Trace>, <Trace>, ...]`</td></tr><tr><td><*Trace*></td><td>String</td><td>A specific trace that the user wishes to delete</td></tr></table>
Return Value	None.

Python Syntax	DeleteTraces(<*TraceSelection*>)

Python Example	`oModule.DeleteTraces (["XY Plot 1:=",`
	`["dB(S(LumpPort1,LumpPort1))"],`
	`"XY Plot 2:=", ["Mag_E"]])`

DoesSupportTraceCharacteristics

Determines whether trace characteristics is supported in a specified display type.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*DisplayType*></td><td>String</td><td>Specified display type to check.</td></tr></table>
Return Value	Integer

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• **1** - trace characteristics is supported. • **0** - trace characteristics not supported.

Python Syntax	DoesSupportTraceCharacteristics(<*DisplayType*>)
Python Example	`oModule.DoesSupportTraceCharacteristics("Rectangular Plot")`

EditCartesianXMarker

Edits an XMarker value.

UI Access	N/A		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of report.
	<*MarkerName*>	String	Marker name, including any trailing number.
	<*XValue*>	Double	X coordinate.
Return Value	None.		

Python Syntax	EditCartesianXMarker (<*ReportName*>, <*MarkerName*>, <*XValue*>)
Python Example	`oModule.EditCartesianXMarker ("XY Plot 1", "MX1", 0)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditCartesianYMarker

Edits a YMarker value.

UI Access	N/A
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report.
<*MarkerName*>	String	Marker name, including any trailing number.
<*AxisName*>	String	Name of axis.
<*YValue*>	Double	Y coordinate.
<*CurveName*>	String	Name of curve.

Return Value	None

Python Syntax	EditCartesianYMarker (<*ReportName*>, <*MarkerName*>, <*AxisName*>, <*YValue*>, <*CurveName*>)
Python Example	`oModule.EditCartesianYMarker("XY Plot 1", "MY1", "Y1", 0, "dB() : Setup1 : Sweep1")`

EditMarker

Edits a marker on a report.

UI Access	N/A
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report.
<*MarkerName*>	String	Marker name, including any trailing number.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*CurveName*>	String	Full trace name.
	<*PrimarySweepValue*>	String	Primary sweep value, including unit.
Return Value	None.		

Python Syntax	EditMarker(<*ReportName*>, <*MarkerName*>, <*CurveName*>, <*PrimarySweepValue*>)
Python Example	`oModule.EditMarker("XY Plot 1", "m5",` `"GS1.VAL : TR4 : Cartesian", "3.61599999999997s")`

ExportEyeMaskViolation

Exports eye mask violations to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of specified report.
	<*ExportFileName*>	String	File name to export to.
Return Value	N/A		

Python Syntax	ExportEyeMaskViolation(<*ReportName*>, <*ExportFileName*>)
Python Example	`oModule.ExportEyeMaskViolation("Variables Plot 1", "C:/eyemask1.csv")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportImageToFile [Reporter]

Exports a report image in a specified format.This command is fully supports -ng (non-graphical) mode.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ReportName></td><td>String</td><td>Name of report to be exported.</td></tr>
<tr><td><FileName></td><td>String</td><td>Full path of the exported image file name; with extension of jpg, gif, tiff, tif, bmp, or wrl.</td></tr>
<tr><td><Width></td><td>Integer</td><td>Image width in pixels; if width or height is less or equal to zero, use the report window width, or 500 pixels if report window is closed.</td></tr>
<tr><td><Height></td><td>Integer</td><td>Image height in pixels; if width or height is less or equal to zero, use the report window height, or 500 pixels report window is closed.</td></tr>
<tr><td><SaveImageParams>> (2D)</td><td>Array</td><td>Whether to ShowLegend, ShowMarkerTable, ShowDeltaMarkerTable. Values can be True, False or Default. Default is True.</td></tr>
<tr><td><SaveImageParams> (3D)</td><td>Array</td><td>Whether to AutoFit [Default is "False"]
ShowLegend, ShowMarkerTable, ShowDeltaMarkerTable. Values can be True, False or Default. Default is True.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	ExportImageToFile(<*ReportName*>, <*FileName*>, <*Width*>, <*Height*>)

Python Example	#### Script Recorded for Saving 2D
	`oDesktop.OpenProject("E:/Ansoft/Tee2.aedt")`
	`oProject = oDesktop.SetActiveProject("Tee2")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oDesign = oProject.SetActiveDesign("TeeModel")

oModule = oDesign.GetModule("ReportSetup")

oModule.ExportImageToFile("S Parameter Plot 1", "E:/Ansoft/S Parameter Plot 1.png",
1920, 1080)

oModule.ExportImageToFile("S Parameter Plot 12", "E:/Ansoft/S Parameter Plot 12.png",
1920, 1080)

oModule.ExportImageToFile("S Parameter Plot 1", "E:/Ansoft/S Parameter Plot 1.png",
1920, 1080,

  [

    "NAME:SaveImageParams",

    "ShowLegend:=" , "False",

    "ShowMarkerTable:=" , "False"

    "ShowDeltaMarkerTable:=" , "False"

  ])

oModule.ExportImageToFile("S Parameter Plot 1", "E:/Ansoft/S Parameter Plot 1.png",
1920, 1080,

  [

    "NAME:SaveImageParams",

    "ShowLegend:=" , "True",

    "ShowMarkerTable:=" , "True"
    "ShowDeltaMarkerTable:=" , "False"

  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule.ExportImageToFile("S Parameter Plot 12", "E:/Ansoft/S Parameter Plot 12.png",
1920, 1080,
  [
    "NAME:SaveImageParams",
    "ShowLegend:=" , "False",
    "ShowMarkerTable:=" , "Default"
    "ShowDeltaMarkerTable:=" , "False"
  ])
oModule.ExportImageToFile("S Parameter Plot 12", "E:/Ansoft/S Parameter Plot 12.png",
1920, 1080,
  [
    "NAME:SaveImageParams",
    "ShowLegend:=" , "True",
    "ShowMarkerTable:=" , "Default"
    "ShowDeltaMarkerTable:=" , "True"
  ])
  oDesktop.CloseProject("Tee2")
```

Script Recorded for saving as 3D:

```
oDesktop.OpenProject("E:/Ansoft/Tee2.aedt")
oProject = oDesktop.SetActiveProject("Tee2")
oDesign = oProject.SetActiveDesign("TeeModel")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule = oDesign.GetModule("ReportSetup")

oModule.ExportImageToFile("S Parameter Plot 6", "E:/Ansoft/S Parameter Plot 11.png",
1335, 466,

  [

    "NAME:SaveImageParams",

    "AutoFit:=" , "False",

    "Orientation:=" , "",

    "ShowOrientationGadget:=", "Default"

  ])

oModule.ExportImageToFile("S Parameter Plot 6", "E:/Ansoft/S Parameter Plot 11.png",
1335, 466,

  [

    "NAME:SaveImageParams",

    "AutoFit:=" , "False",

    "Orientation:=" , "",

    "ShowOrientationGadget:=", "Default"

    "ShowLegend:=" , "True"

  ])

oModule.ExportImageToFile("S Parameter Plot 6", "E:/Ansoft/S Parameter Plot 10.png",
1335, 466,

  [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NAME:SaveImageParams",
    "AutoFit:=" , "False",
    "Orientation:=" , "",
    "ShowOrientationGadget:=", "Default",
    "ShowLegend:=" , "False"
  ])
```

ExportModelImageToFile

Exports the model as an image file (*.avz, *.bmp, *.gif, *.jpeg, *.tiff, *.wrl). In Release 23.1, this command is fully supports -ng (non-graphical) mode. To export to Ensight use *.avz. For export to Ensight in -ng mode, the corresponding version of Ensight must be installed. On Linux, it might need manual set environment variable AWP_ROOT212 to its installation path, e.g. AWP_ROOT212-2=/installations/ansys_inc/v212/ for AnsysEDT v21.2 and Ensight 21.2.

ExportModelImageToFile exports a model image with background type and color that respect the AEDT color scheme by default. You can specify the background type and color with following parameters:

Parameter Name	Description	Parameter Value
BackgroundType	Choose one out of four types of background	Default Plain LinearGradient RadialGradient
BackgroundColor	Plain: Background color LinearGradient/RadialGradient: Background start color	3 integers with a range of [0,255] that represent red, green, and blue respectively
BackgroundContrastColor	Plain: Ignored LinearGradient/RadialGradient: Background end color	3 integers with a range of [0,255] that represent red, green, and blue respectively

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note: Current scripts will not be affected, the image will be exported with default background color as it always does

If no BackgroundType is specified, or BackgroundType is Default, BackgroundColor and BackgroundContrastColor will be ignored, and the image will be exported with default background color

ExportModelImageToFile supports export overlay of polar plot 3D with existing transformation (scaling, rotation and translation) in -ng (non-graphical) mode.

<table>
<tr><td>UI Access</td><td>Modeler > Export.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><path></td><td>String</td><td>Full file path including extension.</td></tr>
<tr><td><width></td><td>Integer</td><td>Width in pixels (use 0 for default).</td></tr>
<tr><td><height></td><td>Integer</td><td>Height in pixels (use 0 for default).</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<pre>["NAME:SaveImageParams",
  "ShowAxis:=" , <string containing boolean>,
  "ShowGrid:=" , <string containing boolean>,
  "ShowRuler:=" , <string containing boolean>,
  "ShowRegion:=" , <string>,
  "Selections:=" , <string>,
  "FieldPlotSelections:=", <string>' # Comma delimited string. #Use to set which field plot to show.</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"FitToSelections:=" , "", **Note:** "FitToSelections" specify geometry objects for the "Fit" operation. "FitToFieldPlotSelection:=" , "" **Note:** "FitToFieldPlotSelections" specifies field plots for the "Fit" operation. "AutoFit:=" , "True", **Note:** If FitToSlections or FitToFieldPlotSelections are used , then AutoFit is True, it makes sure color key does not overlap field plot. It is False by default. If neither is used, then "AutoFit" will "Fit" to full model. "Orientation:=" , <string> "ShowOrientationGadget:=" , <False>]
Return Value	None.		

Python Syntax	ExportModelImageToFile(<*path*> <*width*> <*height*> <*Parameters*>)
Python Example	oEditor.ExportModelImageToFile ("D:/Image.png", 1920, 1080,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:SaveImageParams",
    "ShowAxis:=" , "True",
    "ShowGrid:=" , "True",
    "ShowRuler:=" , "True",
    "ShowRegion:=" , "Default",
    "Selections:=" , "",
    "FieldPlotSelections:=" , "",
    "FitToSelections:=" , "",
    "FitToFieldPlotSelections:=", "",
    "AutoFit:=" , "True",
    "Orientation:=" , ""
])
```

ExportModelMeshToFile

Exports geometry model to a 3D model file (e.g. *.obj, *.wrl, etc.).

UI Access	N/A		
Parameters	Name	Type	Description
	<filePath>	String	Full file path, including extension *.obj, *.wrl, etc
	<selections>	Array	Selected parts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportModelMeshToFile <*filePath*>, <*selections*>)
Python Example	`oEditor.ExportModelMeshToFile("E:/MyDir/scriptRun/2Selected-ng.obj", ["BotCover-","AveragingVolumeAtPeakRMSEfieldLocation"])`

ExportPlot3DToFile [Reporter]

Use: Exports 3D polar, spherical and rectangular plot to a case file. It works in both graphical and NG mode..

Command: None.

Syntax: ExportPlot3DToFile(<plotName>, <path>)

Return Value: A 3D plot file.

Parameters: `<plotName>`

Type: <string>

Plot name.

`<Path>`

Type: <string>

Path to file.

Python Syntax	ExportPlot3DToFile(<*name*> <*Path*>)
Python Example	`oModule = oDesign.GetModule("ReportSetup")` `oModule.UpdateReports(["rE Plot 1"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	oModule.UpdateReports(["Directivity Plot 1"]) oModule.UpdateReports(["Gain Plot 1"]) oModule.ExportPlot3DToFile("rE Plot 1", "D:/projects/test2-output/rEPlot1.case") oModule.ExportPlot3DToFile("Directivity Plot 1", "D:/projects/test2-output/DirectivityPlot1.case") oModule.ExportPlot3DToFile("Gain Plot 1", "D:/projects/test2-output/GainPlot1.case")

ExportReport

Exports a report to a data file.

> **Note:**
>
> The ExportReport script command has been replaced by the script command ExportToFile. ExportReport remains in order to retain backward compatibility for existing scripts, but it is strongly recommended that you now use ExportToFile.

UI Access	None		
Parameters	Name	Type	Description
	<ReportName>	String	Name of report to be exported
	<FileName>	String	Full path of the exported report file name (with extension)
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportReport(<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ExportReport(` `"Plot1", "c:\report1.dat")`

ExportReportDataToFile

Exports report data to a file.

UI Access	N/A
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of specified report.
<*ExportFileName*>	String	Name of export file. File extenstion "rdat" is expected.

Return Value	None.

Python Syntax	ExportReportDataToFile(<*ReportName*>, <*ExportFileName*>)
Python Example	`oModule.ExportReportDataToFile("Plot 1", "C:/Plot1data.rdat")`

ExportTableToFile

Exports a marker table from a report to a file.

UI Access	Right-click on a plot, select **Marker** > **Export Marker Table...** or **Export Delta Marker Table...**
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of specified report.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ExportFileName>	String	Name of export file.
	<TableType>	String	Type of marker table to export. "Marker" or "DeltaMarker".
Return Value	None.		

Python Syntax	ExportTableToFile(*<ReportName>*, *<ExportFileName>*, *<TableType>*)
Python Example	`oModule.ExportTableToFile("Plot 1", "C:/Marker.csv", "Marker")`

ExportToFile [Reporter]

From a data table or plot, generates text format, comma delimited, tab delimited, .dat, or .rdat type output files.

UI Access	Right-click on report name in the Project tree and select **Export**.		
Parameters	Name	Type	Description
	<ReportName>	String	Name of the report
	<FileName>	String	Path and File Name Supported formats

.txt	Post processor format file
.csv	Comma-delimited data file
.tab	Tab-separated file
.dat	Ansys plot data file
.rdat	Ansys report data file

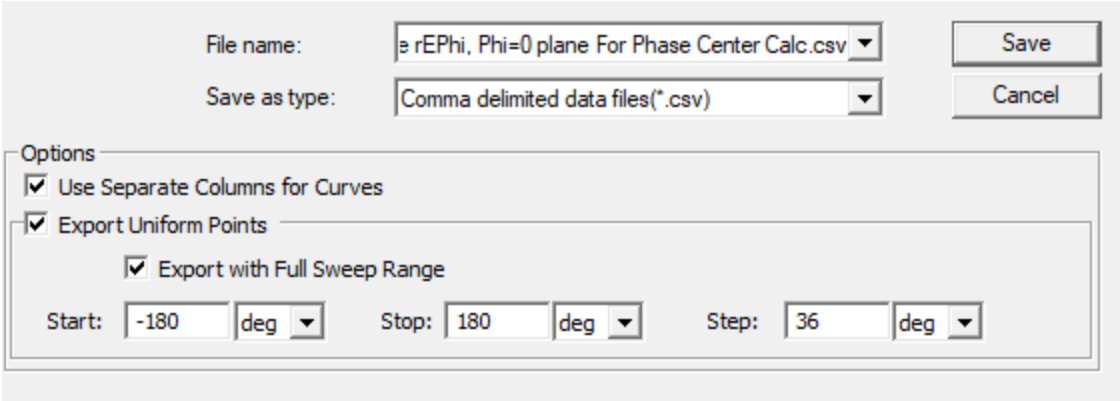
Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	ExportToFile (<*ReportName*>, <*FileName*>)
Python Example	`oModule.ExportToFile(` `"Plot1", "c:\report1.dat")`

ExportUniformPointsToFile

Exports uniform points from a data table or plot report that includes the Export Uniform Points to File option enabled to text format, comma delimited, tab delimited, or .dat type output files.



UI Access	Right-click on report name in the project tree and select **Export Data**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report to be exported.
<*FileName*>	String	Full path of the exported image file name; with extension of .txt - Post processor format file .csv - Comma-delimited data file .tab - Tab-separated file .dat - Ansys plot data file
<SweepRange>	String	Start, stop, and step range with units, for sweep.
<UnitSpec>	String	For example, “kV, Mhz, yd”
<UseTraceNumberFormat>	Boolean	"True", "False"

Return Value	None.

Python Syntax	ExportUniformPointsToFile(<*ReportName*>, <*FileName*>, <*SweepRange*>)
Python Example	`oModule.ExportUniformPointsToFile("S Parameter Table 2", "D:/MyFiles/cftt.csv", "4GHz", "5GHz", "1GHz", False, " kV, MHz, yd ", False)`

GetAllCategories

Get all available category names (not including variable and output-variables) in a solution for a report type and display type, returned as an array of text strings.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportType*></td><td>String</td><td>Report type name.</td></tr><tr><td><*DisplayType*></td><td>String</td><td>Name of display type.</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of solution.</td></tr><tr><td><*SimValueCtxt*></td><td>Array</td><td>A context name or array of strings that encode the contexts.</td></tr></table>
Return Value	Array of text strings

Python Syntax	GetAllCategories(<*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*SimValueCtxt*>)
Python Example	`oModule.GetAllCategories("Far Fields",` `"Rectangular Plot", "Setup1 : LastAdaptive", "Infinite Sphere1")`

GetAllQuantities

Gets all available quantity names in category, returned as an array of text strings.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportType*></td><td>String</td><td>Report type name.</td></tr><tr><td><*DisplayType*></td><td>String</td><td>Display type name.</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of solution.</td></tr><tr><td><*SimValueCtxt*></td><td>Array</td><td>A context name, or array of strings that encoded the contexts(I).</td></tr><tr><td><*CategoryName*></td><td>String</td><td>A category name as input parameter. a category name returned in GetAllCategories() or "Variables", or "Output Variables"</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Array of text strings

Python Syntax	GetAllQuantities(<*ReportType*>,<*DisplayType*>, <*SolutionName*>, <*SimValueCtxt*>, <*CategoryName*>)

Python Example	`oModule.GetAllQuantities(`
	`  "Far Fields", "Rectangular Plot",`
	`  "Setup1 : LastAdaptive",`
	`  "Infinite Sphere1", "Gain")`

GetAllReportNames

Gets the names of existing reports in a design

UI Access	N/A
Parameters	None.
Return Value	Array of report names

Python Syntax	GetAllReportNames()
Python Example	`oModule.GetAllReportNames()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAvailableDisplayTypes

Retrieves all supported display types in report type as an array of text strings.

UI Access	N/A
Parameters	Name | Type | Description <*ReportType*> | String | Report type name.
Return Value	Array of text strings

Python Syntax	GetAvailableDisplayTypes(<*ReportType*>)
Python Example	`oModule.GetAvailableDisplayTypes("Far Fields")`

GetAvailableReportTypes

Retrieves all available report types in the current Design as an array of text string.

UI Access	N/A
Parameters	None.
Return Value	Array of text strings

Python Syntax	GetAvailableReportTypes()
Python Example	`oModule.GetAvailableReportTypes()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAvailableSolutions

Gets all available solutions in report type as an array of text strings.

UI Access	N/A
Parameters	Name | Type | Description <*ReportType*> | String | Report type name.
Return Value	Array of text strings

Python Syntax	GetAvailableSolutions(<*ReportType*>)
Python Example	`oModule.GetAvailableSolutions("Far Fields")`

GetChildNames [Report Setup]

Gets a list of report names.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing all report names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetChildNames()
Python Example	`oModule.GetChildNames()`

GetChildObject [Report Setup]

Gets a report object or report child object; the module's first level of child object is report. Report has trace, axis, header, Legend, and more children. Trace has curve as child etc. Those child objects can be accessed by calling all levels of parent object's GetChildObject(path) function.

UI Access	NA
Parameters	Name | Type | Description <*ObjectPath*> | String | Report Name or an object path beginning with a report name.
Return Value	A ReportSetup(Results) Module Child Object, [ReportSetup(Results) Module Child Objects]

Python Syntax	GetChildObject(<*ObjectPath*>)

Python Example	`oRpt = oRptModule.GetChildObject("S Parameter Plot 1")`
	`oTrace = oRptModule.GetChildObject("S Parameter Plot 1/dB(S(Port1,Port1))")`
	`oAxisX = oRptModule.GetChildObject("S Parameter Plot 1/AxisX")`

GetChildTypes [ReportSetup]

Gets child types of queried Report module object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing child object types.

Python Syntax	GetChildTypes ()
Python Example	`oModule.GetChildTypes()`

GetCurvePropServerName

Gets the PropServer (the owner of the properties, or the list containing them) name of a curve.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of specified report. <*TraceName*> | String | Name of specified trace.
Return Value	Array of string containing the PropServer name.

Python Syntax	GetCurvePropServerName(<*ReportName*>, <*TraceName*>)
Python Example	`oModule.GetCurvePropServerName("Plot 1", "Phase")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDisplayType

Gets the display type of a specified report.

UI Access	NA
Parameters	Name | Type | Description <*ReportName*> | String | Report name
Return Value	String containing display type.

Python Syntax	GetDisplayType(<*ReportName*>)

Python Example	`oDesign = Project.SetActiveDesign("Design1")`
	`oModule = oDesign.GetModule("ReportSetup")`
	`oModule.GetDisplayType("Design Plot 1")`

GetDynLinkIntrinsicVariables

Gets variable names from a trace included in dynamic link outputs.

UI Access	N/A
Parameters	Name | Type | Description <*TraceName*> | String | Trace name with its report name.
Return Value	Array of variable names

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDynLinkIntrinsicVariables(<*TraceName*>)
Python Example	`oModule.GetDynLinkIntrinsicVariables("Plot 1:Trace")`

GetDynLinkQtyValueState

Gets the state of the quantity values from a source dynamic linked trace.

UI Access	N/A
Parameters	Name | Type | Description <*TraceName*> | String | Name of specified source trace. <*QtyName*> | String | Name of specified quantity value.
Return Value	Array of strings containing states.

Python Syntax	GetDynLinkQtyValueState(<*TraceName*>, <*QtyName*>)
Python Example	`oModule.GetDynLinkQtyValueState("Plot 1:Trace1", "")`

GetDynLinkTraces

Gets the names of the dynamic linked traces.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name | Type | Description <*SoluName*> | String | Name of the source solution. If empty, refer to current solution.
Return Value	Array of strings containing trace names.

Python Syntax	GetDynLinkTraces(<*SoluName*>)
Python Example	`oModule.GetDynLinkTraces("")`

GetDynLinkVariableValues

Gets the values of a variable from a dynamic linked trace.

UI Access	N/A
Parameters	Name | Type | Description <*TraceName*> | String | Name of specified dynamic linked trace, with its report name. <*VarName*> | String | Name of specified variable.
Return Value	Array of strings containing variable values.

Python Syntax	GetDynLinkVariableValues(<*TraceName*>, <*VarName*>)
Python Example	`oModule.GetDynLinkVariableValues("Plot 1:Trace", "Var1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetObjPath [Design]

Obtains the path to the design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the design.

Python Syntax	GetObjPath()
Python Example	`oDesign.GetObjPath()`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

> **Tip:** Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PropTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<PropServer>	String	An object identifier, generally returned from another script method, such as `Comp‡nst@R;2;3`
	<PropName>	String	Name of the property.
Return Value	String value of the property.		

Python Syntax	GetPropertyValue (*<PropTab>*, *<PropServer>*, *<PropName>*)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`
	`...`

GetPropNames [Reporter]

Report setup module does not have its own property, this function always returns empty array.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	Empty array.

Python Syntax	GetPropNames()
Python Example	`oModule.GetPropNames()`

GetPropValue [Report Setup]

Gets the property value for a Report, or reports' child object.

UI Access	N/A
Parameters	Name | Type | Description *<PropPath>* | String | A child object's property path. See property path discussion here.
Return Value	String containing the property value.

Python Syntax	GetPropValue(*<PropPath>*)
Python Example	`oModule.GetPropValue("S Parameter Plot 1/Display Type")`

GetQtyExpressionsForSourceTrace

Gets the quantity expressions from a specified source trace.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	Name | Type | Description <*SourceTraceName*> | String | Name of specified source trace.
Return Value	Array of strings containing quantity expressions.

Python Syntax	GetQtyExpressionsForSourceTrace(<*SourceTraceName*>)
Python Example	`oModule.GetQtyExpressionsForSourceTrace("Plot 1:Trace1")`

GetReportTraceNames

Gets the names of existing trace names in a plot.

UI Access	N/A
Parameters	Name | Type | Description <*PlotName*> | String | Name of specified plot.
Return Value	Array of strings containing trace names.

Python Syntax	GetReportTraceNames(<*PlotName*>)
Python Example	`oModule.GetReportTraceNames("SParameter Plot 1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetReportSummaryForRegressionTesting

Gets report summary from a dumped report file.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of a specified dumped report.</td></tr></table>
Return Value	String containing report summary.

Python Syntax	GetReportSummaryForRegressionTesting(<*ReportName*>)
Python Example	`oModule.GetReportSummaryForRegressionTesting("C:/report.rdat")`

GetSolutionContexts

Gets all available solution context names in a solution as an array of text strings.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportType*></td><td>String</td><td>Report type name.</td></tr><tr><td><*DisplayType*></td><td>String</td><td>Display type name.</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of solution.</td></tr></table>
Return Value	Array of text strings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetSolutionContexts(<*ReportType*>, <*DisplayType*>, <*SolutionName*>)
Python Example	`oModule.GetSolutionContexts(` `"Far Fields", "Rectangular Plot", "Setup1:LastAdaptive")`

GetSolutionDataPerVariation

Obtains solution data for a given report type and solution. You must have already run a simulation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><reportTypeArg></td><td>String</td><td>Report type name as input parameter.</td></tr><tr><td><solutionNameArg></td><td>String</td><td>Solution name as input parameter.</td></tr><tr><td><simValueCtxtArg></td><td>Structured Array</td><td>Same as ContextArray values created in the relevant CreateReport script.</td></tr><tr><td><familiesArg></td><td>Array of Strings</td><td>Same as FamiliesArray values created in the relevant CreateReport script.</td></tr><tr><td><expressionArg></td><td>String or Array of Strings</td><td>Text string or array of text strings; valid expression, may validate it as the data-table Y-component.</td></tr></table>
Return Value	ARRAY of ISolutionDataResultComInterface objects, containing: • GetSweepNames() • GetSweepUnits() • GetSweepValues() • IsPerQuantityPrimarySweep() • GetPerQuantityPrimarySweepValues()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• IsDataComplex() • GetDataUnits() • GetRealDataValues() • GetImagDataValues() • ReleaseData() • GetDesignVariableNames() • GetDesignVariableUnits() • GetDesignVariableValue() • GetDesignVariationKey()

Note:

- This command is *not* recordable from the UI, but its parameters are similar to CreateReport, so you may record a CreateReport script to get the parameter values.
- For the returned ISolutionDataResultComInterface object, some of its functions have an optional boolean parameter: SIValue. SIValue defaults to True. When the pass in value is True, return data values will be in Standard International values; when False, return data values will be in the current units.

  Example: Freq Sweep with [1GHz, 2GHz,3GHz], GetSweepUnits("Freq") return "GHz"; GetSweepValues("Freq", True) return [1000000,2000000,3000000]; GetSweepValues("Freq", False) return [1,2,3].

Python Syntax	GetSolutionDataPerVariation(reportTypeArg, solutionNameArg, simValueCtxtArg, familiesArg, expressionArg)

Python Example	`oModule = oDesign.GetModule("ReportSetup")`
	
	`arr = oModule.GetSolutionDataPerVariation('Modal Solution Data', 'Setup1 : Sweep', ['Domain:=', 'Sweep'], ['Freq:=', ['All'], 'offset:=', ['All']], ['S(Port1,Port1)', 'dB(S(Port1,Port3))'])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDataUnits

Returns text string containing units.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	Name: <expressionString> — Type: String — Description: Can be returned from `GetDataExpressions()`
Return Value	Text string of units; empty if no units

Python Syntax	GetDataUnits(<expressionString>)
Python Example	`oModule.GetDataUnits(expressions)`

GetDesignVariableNames

Returns array of strings containing design variable names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	NA
Parameters	NA
Return Value	Array of strings

Python Syntax	GetDesignVariableNames()
Python Example	`names = oModule.GetDesignVariableNames()`

GetDesignVariableUnits

Returns array of strings containing design variable units.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<varName>	String	Can be returned from GetDesignVariableNames()
Return Value	Text string of units; empty if no units.		

Python Syntax	GetDesignVariableUnits(<varName>)
Python Example	`units = oModule.GetDesignVariableUnits('Variable Name')`

GetDesignVariableValue

Returns a design variable's value.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A		
Parameters	Name	Type	Description
	<varName>	String	Can be returned from GetDesignVariableNames()
	<siValue>	Boolean	True to return SI-value; False to return by GetDesignVariableUnits()
Return Value	Double value		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDesignVariableValue(<varName>, <siValue>)
Python Example	`value = oModule.GetDesignVariableValue('varName',1)`

GetDesignVariationKey

Returns a design's Variation Key.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	String containing variation key.

Python Syntax	GetDesignVariationKey()
Python Example	`oModule.GetDesignVariationKey()`

GetImagDataValues

Returns array of imaginary data values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`</td></tr><tr><td><siValue></td><td>Integer</td><td>1 to return SI-value; 0 to return with units returned in GetSweepUnits().</td></tr></table>
Return Value	Array of doubles

Python Syntax	GetImagDataValues(<expressionString>,<siValue>)
Python Example	`imaginaryvalues = oModule.GetImagDataValues('expression',1)`

GetPerQuantityPrimarySweepValues

Returns per quantity primary sweep values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`</td></tr><tr><td><siValue></td><td>Integer</td><td>1 to return SI-value; 0 to return by GetSweepUnits().</td></tr></table>
Return Value	Array of doubles if IsPerQuantityPrimarySweep() returned True; error if returned False

Python Syntax	GetPerQuantitySweepValues(<expressionString>, <siValue>)
Python Example	`sweepvalues = oModule.GetPerQuantitySweepValues('0.111,0.201,0.345,0.231', 1)`

GetRealDataValues

Returns array of real data values.

Important:

This is a member function of `ISolutionDataResultComInterface` object, which is the element of the returned array from function GetSolutionDataPerVariation.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`</td></tr><tr><td><siValue></td><td>Integer</td><td>1 to return SI-value; 0 to return with units returned in GetSweepUnits().</td></tr></table>
Return Value	Array of doubles

Python Syntax	GetRealDataValues(<expressionString>,<siValue>)
Python Example	`realvalues = oModule.GetRealDataValues('expression',1)`

GetSweepNames

Returns array of text strings containing primary sweep name(s).

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	Array of text strings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetSweepNames()
Python Example	`sweepnames = arr[0].GetSweepNames()`

GetSweepUnits

Returns text string containing units.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	Name: <sweepName> — Type: String — Description: Primary sweep name
Return Value	Text string containing units

Python Syntax	GetSweepUnits(<sweepName>)
Python Example	`sweepunits = oModule.GetSweepUnits('Sweep 1')`

GetSweepValues

Returns sweep values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><sweepName></td><td>String</td><td>Primary sweep name</td></tr><tr><td><siValue></td><td>Boolean</td><td>True to return SI-value; False to return by GetSweepUnits().</td></tr></table>
Return Value	Array of doubles

Python Syntax	GetSweepValues(<sweepName>, <siValue>)
Python Example	`sweepvalues = oModule.GetSweepValues('Sweep 1', True)`

IsDataComplex

Returns whether an expression is complex.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`.</td></tr></table>
Return Value	Boolean (True if expression is Complex data; False if not)

Python Syntax	IsDataComplex(<expressionString>)
Python Example	`oModule.IsDataComplex('.001,.234,.455,.434')`

IsPerQuantityPrimarySweep

Returns whether data expressions have different primary sweep values.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	N/A
Return Value	Boolean (True if data expressions have different primary sweep values)

Python Syntax	IsPerQuantityPrimarySweep()
Python Example	`var = oModule.IsPerQuantityPrimarySweep()`

Release Data

Releases all cached data. After this function is called, all subsequent function calls to the object will fail.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	N/A

Python Syntax	ReleaseData()
Python Example	`oModule.ReleaseData()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GroupPlotCurvesByGroupingStrategy

Groups curves in a Stacked Plot automatically based on a curve grouping strategy.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of report. <GroupStrategy> | String | Strategy for grouping, "Single", "By Trace" or "By Units".
Return Value	None.

Python Syntax	GroupPlotCurvesByGroupingStrategy(<*ReportName*>, <*GroupStrategy*>)
Python Example	`oModule.GroupPlotCurvesByGroupingStrategy("Transient Plot 1", "By Trace")`

ImportIntoReport

Imports .tab, .csv, and .dat format files into a report.

UI Access	Right-click on report name in the Project tree and select **Import...**.
Parameters	Name | Type | Description <*ReportName*> | String | Name of the Report <*FileName*> | String | Path and File Name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		.csv	Comma-delimited data file
		.tab	Tab-separated file
		.dat	Ansys plot data file

Return Value	None

Python Syntax	ImportIntoReport (<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ImportIntoReport ("Plot1", "c:\report1.dat")`

ImportReportDataIntoReport

Imports report data from a file into a specified report.

UI Access	N/A		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of specified report.
	<*FileName*>	String	Name of specified data file. File extenstion "rdat" is expected.
Return Value	None.		

Python Syntax	ImportReportDataIntoReport(<*ReportName*>, <*FileName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.ImportReportDataIntoReport("Plot 1", "C:/Plot1data.rdat")`

MovePlotCurvesToGroup

In a Stacked Plot move curve(s) from its stack(s) to an existing stack. Here term ‘group’ is synonymous to ‘stack’ in the context of cartesian stacked plot.

UI Access	N/A
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report.
<*CurveArray*>	Array	Array of curve names to move.
<*StackName*>	String	Name of stack to move to.

Return Value	None.

Python Syntax	MovePlotCurvesToGroup(<*ReportName*>, <*CurveArray*>, <*StackName*>)
Python Example	`oModule.MovePlotCurvesToGroup("XY Stacked Plot 1", ["R2.V : TR", "R2.I : TR"], "Stack 2")`

MovePlotCurvesToNewGroup

Move curve(s) from its stack(s) to a new stack. Here term ‘group’ is synonymous to ‘stack’ in the context of Cartesian stacked plot.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report.</td></tr><tr><td><*CurveArray*></td><td>Array</td><td>Array of curve names to move.</td></tr></table>
Return Value	None.

Python Syntax	MovePlotCurvesToNewGroup (<*ReportName*>, <*CurveArray*>)

Python Example	`oModule.MovePlotCurvesToNewGroup(`
	`"XY Stacked Plot 1",`
	`["R2.V : TR", "R2.I : TR"])`

OpenWindowForAllReports

Opens windows for all reports belong to current design.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	OpenWindowForAllReports()
Python Example	`oModule.OpenWindowForAllReports()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

OpenWindowForReports

Opens windows for specified reports.

UI Access	N/A
Parameters	Name | Type | Description *<ReportNames>* | Array | Array of strings containing report names.
Return Value	None.

Python Syntax	OpenWindowForReports(*<ReportNames>*)
Python Example	`oModule.OpenWindowForReports(["Report1", "Report2"])`

PastePlotSettings

Paste plot settings to a specified report.

UI Access	Right-click a report, select **Paste**
Parameters	Name | Type | Description *<ReportName>* | String | Name of specified report. *<PropTypeToApply>* | String | Property type to paste. "Graphical", "Data", or "All".
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	PastePlotSettings(<*ReportName*>, <*PropTypeToApply*>)
Python Example	`oModule.PastePlotSettings("Plot 1", "Graphical")`

PasteReports

Paste copied reports to results in the current project.

UI Access	**Edit** > **Paste**
Parameters	None.
Return Value	None.

Python Syntax	PasteReports ()
Python Example	`oModule.PasteReports()`

PasteReportsWithLegacyNames

Pastes copied reports to results in the current project with legacy name definitions.

UI Access	N/A
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	PasteReportsWithLegacyNames ()
Python Example	`oModule.PasteReportsWithLegacyNames()`

PasteTraces

Pastes copied traces to a named plot.

UI Access	Paste
Parameters	Name | Type | Description <*ReportName*> | String | Name of plot
Return Value	None

Python Syntax	PasteTraces (<*ReportName*>)
Python Example	`oModule.PasteTraces ("XY Plot1")`

PasteTracesWithLegacyNames

Pastes copied traces to a named plot using legacy name definitions.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of plot</td></tr></table>
Return Value	None

Python Syntax	PasteTracesWithLegacyNames (<*ReportName*>)
Python Example	`oModule.PasteTracesWithLegacyNames("XY Plot1")`

RenameReport

Renames an existing report.

UI Access	Select a report on the Project tree, right-click and select **Rename**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OldReportName*></td><td>String</td><td>Old Report Name</td></tr><tr><td><*NewReportName*></td><td>String</td><td>New Report Name</td></tr></table>
Return Value	None.

Python Syntax	RenameReport (<*OldReportName*>, <*NewReportName*>)
Python Example	`oModule.RenameReport("XY Plot1", "Reflection")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RenameTrace

To rename a trace in a plot

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of report. <*TraceName*> | String | Name of Trace <*NewName*> | String | New trace name.
Return Value	None.

Python Syntax	RenameTrace(<*ReportName*>, <*TraceName*>, <*NewName*>)
Python Example	`oModule.RenameTrace ("XY Plot1",` `"dB(S(WavePort1,WavePort1))1", "Port1dbS")`

ResetPlotSettings

Resets plot settings to defaults.

UI Access	Right-click on a plot, select **Edit** > **Reset Plot Settings**
Parameters	Name | Type | Description <*PlotName*> | String | Name of specified plot.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ResetPlotSettings(<*PlotName*>)
Python Example	`oModule.ResetPlotSettings("Differential S-parameters")`

SavePlotSettingsAsDefault

Saves report plot settings as default.

UI Access	**Report Templates** > **Save Settings as Default**
Parameters	Name | Type | Description <*PlotName*> | String | Name of plot to use for plot defaults.
Return Value	None.

Python Syntax	SavePlotSettingsAsDefault ("<*PlotName*>")
Python Example	`oModule.SavePlotSettingsAsDefault("XY Plot1")`

SetLinkOutputTraces

Specifies dynamic link output traces from the current design.

UI Access	Right-click the **Results**, select **Link Output...**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*TraceArray*></td><td>Array</td><td>Array of traces to set. ["<ReportName>:=", <array of trace names>, "<ReportName>:=", <array of trace names>,...]</td></tr></table>
Return Value	None.

Python Syntax	SetLinkOutputTraces(<*TraceArray*>)

Python Example	`oModule.SetLinkOutputTraces(`
	`[`
	`  "Plot 1:=", ["Trace1"],`
	`  "Plot 2:=", ["Trace1"]`
	`])`

SetPropValue [Report Setup]

Sets the property value for report module child object.

UI Access	Select **Edit Properties** on Report objects.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*PropPath*></td><td>String</td><td>A child object's property path. See property path discussion here.</td></tr><tr><td><*NewValue*></td><td>String, Number, or Boolean</td><td>New value data type is depending on the property type,</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SetPropValue(<*PropPath*>, <*NewValue*>)
Python Example	`oRptModule.SetPropValue("S Parameter Plot 1/Display Type", "DataTable")` `oRptModule.SetPropValue("S Parameter Plot 1/db(S(port1,port2)/Primary sweep", "Freq")`

UnGroupPlotCurvesInGroup

From a Stacked Plot, ungroups curves in a stack.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of report. <*GroupName*> | String | Stack group name.
Return Value	None.

Python Syntax	UnGroupPlotCurvesInGroup(<*ReportName*>, <*GroupName*>)
Python Example	`oModule.UngroupPlotCurvesInGroup("S Parameter Plot 3", "Stack 1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateAllReports

Updates all reports in the **Results** branch in the project tree.

UI Access	Right-click on **Results** in the project tree, select **Update All Reports**
Parameters	None
Return Value	None

Python Syntax	UpdateAllReports()
Python Example	`oModule.UpdateAllReports()`

UpdateReports

Updates specified reports.

UI Access	N/A
Parameters	Name | Type | Description <*ReportNames*> | Array | Array of strings containing report names.
Return Value	None.

Python Syntax	UpdateReports(<*ReportNames*>)
Python Example	`oModule.UpdateReports (["XY Plot 1", "XY Plot 4"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateTraces

Update the traces in a report for which traces are not automatically updated by the Report Traces dialog box, Update Report, Real Time selection.

<table>
<tr><td>UI Access</td><td>In Report dialog, click Apply Traces button</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ReportName></td><td>String</td><td>Name of Report.</td></tr>
<tr><td><TraceNames></td><td>Array</td><td>Array of strings containing trace names.</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution.</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Context for which the expression is being evaluated. This can be an empty string if there is no context.
["Domain:=", <DomainType>)
<DomainType> ex. "Sweep" or "Time"
["Context:=", <GeometryType>]
<GeometryType> ex. "Infinite Spheren", "Spheren", "Polylinen"</td></tr>
<tr><td><FamiliesArray></td><td>Array</td><td>Contains sweep definitions for the report.
["<VariableName>:= ", <ValueArray>]
<ValueArray>
["All"] or ["Value1", "Value2", ..."Valuen"]
examples of <VariableName> "Freq", "Theta", "Distance"</td></tr>
<tr><td><ReportDataArray></td><td>Array</td><td>This array contains the report quantity and X, Y, and (Z) axis definitions.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["X Component:=", <VariableName>, "Y Component:=", <VariableName> | <ReportQuantityArray>] <ReportQuantityArray> ex. ["dB(S(Port1, Port1))"]
	<ExtTraceInfo>	Array	Optional. Array defines extended trace information.
Return Value	None.		

Python Syntax	UpdateTraces(*<ReportName>*, *<SolutionName>*, *<ContextArray >*, *<FamiliesArray>*, *<ReportDataArray>*)
Python Example	oModule.UpdateTraces("XY Plot 1", ["NEG1.VAL"], "TR4", ["NAME:Context", "SimValueContext:=", [2,0,2,0,False,False, -1,1,0,1,1,"",0,0,"CG",False,"0","KP",False,"0","MH",False, "100","TE",False,"100s","TH",False,"40", "TS",False,"0ns","UF",False, "0","WT",False,"0","WW",False,"100"]], ["Spectrum:=", ["All"]],

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["X Component:=", "Spectrum", "Y Component:=", ["mag(NEG1.VAL)"]], [])

UpdateTracesContextAndSweeps

Edits sweeps and context of multiple traces without affecting their component expressions.

UI Access	**Modify Report** with multiple traces selected.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of Report.</td></tr><tr><td><*TraceNames*></td><td>Array</td><td>Array of strings containing trace names.</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of the solution as listed in the **Modify Report** dialog box. For example: "Setup1 : Last Adaptive"</td></tr><tr><td><*ContextArray*></td><td>Array</td><td>Context for which the expression is being evaluated. This can be an empty string if there is no context. ex. "Sweep" or "Time"</td></tr><tr><td><*PointSet*></td><td>Array</td><td>Point set for the selected traces, for example, X and Y values for the plot.</td></tr></table>
Return Value	None

Python Syntax	UpdateTracesContextAndSweeps(<*ReportName*>, <*TraceNames*>, <*SolutionName*>, <*ContextArray*>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*PointSet*>)
Python Example	oModule.UpdateTracesContextAndSweeps_ ("Active S Parameter Quick Report", ["dB(ActiveS(Port1:1))", "dB(ActiveS(Port2:1))"], "Setup1 : Sweep1", [], ["Freq:=", ["9GHz", "9.05GHz", "9.1GHz", "9.15GHz", "9.2GHz", "9.25GHz", "9.3GHz", "9.35GHz", "9.4GHz", "9.45GHz", "9.5GHz", "9.55GHz", "9.6GHz", "9.65GHz", "9.7GHz", "9.9GHz", "9.95GHz", "10GHz"], "offset:=",["All"]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

11 - Optimetrics Module Script Commands

Optimetrics script commands should be executed by the `Optimetrics` module.

```
oModule = oDesign.GetModule("Optimetrics")
oModule.CommandName <args>
```

Conventions Used in this Chapter

`<VarName>`

Type: <string>

Name of a variable.

`<VarValue>`

Type: <string>

Value with unit (i.e., <value>, but cannot be an expression).

`<StartV>`

Type: <VarValue>

The starting value of a variable.

`<StopV>`

Type: <VarValue>

The stopping value of a variable.

`<MinV>`

Type: <VarValue>

The minimum value of a variable.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<MaxV>

Type: <VarValue>

The maximum value of a variable.

<IncludeVar>

Type: <bool>

Specifies whether the variable is included in the analysis.

<StartingPoint>

```
Array("NAME:StartingPoint", "<VarName>:=",
  <VarValue>, .... "<VarName>:=", <VarValue>)
```

<SaveField>

Type: <bool>

Specifies whether HFSS will remove the non-nominal field solution.

<MaxIter>

Type: <int>

Maximum iteration allowed in an analysis.

<PriorSetup>

Type: <string>

The name of the embedded parametric setup.

<Precede>

Type: <bool>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

If true, the embedded parametric setup will be solved before the analysis begins.

If false, the embedded parametric setup will be solved during each iteration of the analysis.

```
<Constraint>
  Array("NAME:LCS",
    "lc:=", Array("<VarName>:=",
      <Coeff>, …"<VarName>:=", <Coeff>, "rel:=", <Cond>, "rhs:=", <Rhs>), ...
    "lc:=", Array("<VarName>:=", <Coeff>, ..."
      <VarName>:=", <Coeff>, "rel:=", <Cond>, "rhs:=",
      <Rhs>))
```

<Coeff>

Type: <double>

Coefficient for a variable in the linear constraint.

<Cond>

Type: <string>

Inequality condition.

<Rhs>

Type: <double>

Inequality value.

<OptiGoalSpec>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Solution:=", <Soln>, "Calculation:=", <Calc>,
   "Context:=, <Geometry>
    Array("NAME:Ranges",
     "Range:", Array("Var:=",
       <VarName>, "Type:=", <RangeType>, "Start:=",
       <StartV>, "Stop:=", <StopV>), ...
     "Range:", Array("Var:=", <VarName>, "Type:=",
       <RangeType>, "Start:=", <StartV>, "Stop:=",
       <StopV>))
```

<Soln>

Type: <string>

Name of the solution.

<Calc>

Type: <string>

An expression that is composed of a basic solution quantity and an output variable.

<ContextName>

Type: <string>

Name of context needed in the evaluation of <Calc>.

<Geometry>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: <string>

Name of geometry needed in the evaluation of <Calc>.

`<RangeType>`

Type: <string>

if "r", start and stop values specify a range for the variable.

if "s", start values specify the single value for the variable.

The commands in this section include:

CopySetup

DeleteSetups [Optimetrics]

DistrbutedAnalyzeSetup

EditSetup

EditSetup [Parametric]

EditSetup [Optimization]

EditSetup [Sensitivity]

EditSetup [Statistical]

EnableSetup

ExportDXConfigFile

ExportOptimetricsProfile

ExportOptimetricsResult

ExportParametricResults

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportParametricSetupTable

ExportRespSurfaceMinMaxTable

ExportRespSurfaceRefinePoints

ExportRespSurfaceResponsePoints

ExportRespSurfaceVerificationPoints

GenerateVariationData [Parametric]

GetChildNames [Optimetrics]

GetChildObject [Optimetrics]

GetChildTypes [Optimetrics]

GetName

GetObjPath [Editor]

GetOptimetricResult

GetPropNames [Optimetrics]

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

ImportSetup

InsertSetup

InsertSetup [Parametric]

InsertSetup [Optimization]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

InsertSetup [Sensitivity]

InsertSetup [Statistical]

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

SolveAllSetup

SolveSetup [Optimetrics]

The topics for this section include:

General Commands Recognized by the Optimetrics Module

Parametric Script Commands

Optimization Script Commands

Sensitivity Script Commands

Statistical Script Commands

CopySetup

Copy the specified Optimetrics setup.

UI Access	NA
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CopySetup (<*SetupName*>)
Python Example	`oModule.CopySetup ("OptimizationSetup1")`

DeleteSetups [Optimetrics]

Deletes the specified Optimetrics setups.

UI Access	Right-click the setup in the project tree, and then click **Delete** on the shortcut menu
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><NameArray></td><td>Array of Strings</td><td>An Array of Setup Names</td></tr></table>
Return Value	None

Python Syntax	DeleteSetups (<NameArray>)
Python Example	`oModule.DeleteSetups (["OptimizationSetup1"])`

DistributedAnalyzeSetup

Distributes all variable value instances within a parametric sweep to different machines already specified from within the user interface

UI Access	Right-click the parametric setup name in the project tree and select Distribute Analysis.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ParametricSetupName>	String	Name of the Setup
	<isBlocking>	Boolean	An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None		

Python Syntax	DistributedAnalyzeSetup (*<ParametricSetupName>*)
Python Example	`oModule.DistributedAnalyzeSetup("ParametricSetup1")`

EnableSetup

Enables and disables a defined optimetrics analysis setup.

UI Access	Right-click on a setup in the project tree, select **Enable Setup** or **Disable Setup**		
Parameters	Name	Type	Description
	<SetupName>	String	Name of specified setup.
	<Enable>	Boolean	Determines whether enable or disable a setup. • **True** - enable setup. • **False** - disable setup.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnableSetup(<*SetupName*>, <*Enable*>)
Python Example	`oModule.EnableSetup("OptimizationSetup1", True)`

ExportDXConfigFile

Create an xml file with the setup information for Design Xplorer

UI Access	Right click on the Design Xplorer setup in the project tree and choose **Export External Connector Addin Configuration…**
Parameters	Name / Type / Description <SetupName> / String / Must be one of existing DesignExplorer setup names <FileName> / String / Must be a valid file path and name
Return Value	None

Python Syntax	ExportDXConfigFile (<SetupName>, <FileName>)
Python Example	`oModule.ExportDXConfigFile ("DesignXplorerSetup1",` `"c:/exportdir/DXSetup1.xml")`

ExportOptimetricsProfile

Export Optimetrics profile data

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right click on the Optimetrics setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Profile** tab and click on the **Export** button.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Parametric, Optimization, Sensitivity, Statistical or DesignXplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat, or txt</td></tr><tr><td>[profileNum]</td><td>String</td><td>Must be a numeric string. Optional: defaulted to last profile number. It should be a zero indexed profile number.</td></tr></table>
Return Value	None

Python Syntax	ExportOptimetricsProfile (<SetupName>, <FileName>, [profileNum]
Python Example	`oModule.ExportOptimetricsProfile ("StatisticalSetup1", "c:/exportdir/test.csv")`

ExportOptimetricsResult

Export an existing Optimization, Sensitivity, Statistical or DesignXplorer result. (Does not export Parametric results.)

UI Access	Right click on the desired Optimetrics setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Optimization, Sensitivity, Statistical, or DesignXplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat, or txt..</td></tr><tr><td>[useFullOutputName]</td><td>Boolean</td><td>Optional: defaulted to false. If set to true values will be printed with units. This parameter is ignored for Optimization and Statistical results.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	ExportOptimetricsResult (<SetupName>, <FileName>, [useFullOutputName])
Python Example	`oModule.ExportOptimetricsResult (` `"StatisticalSetup1", "c:/exportdir/test.csv",false)`

ExportParametricResults

Export existing Parametric results.

UI Access	Right click on the desired Parametric setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Parametric setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat or txt</td></tr><tr><td><bOutputUnits></td><td>Boolean</td><td>If set to true, values will be printed with units</td></tr></table>
Return Value	None

Python Syntax	ExportParametricResults (<SetupName>, <FileName>, <bOutputUnits>)
Python Example	`oModule.ExportParametricResults(` `"ParametricSetup1", "c:/exportdir/test.csv", False)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportParametricSetupTable

Exports the parametric setup table as a CSV file.

UI Access	Double-click parametric setup. Select **Table** tab. Click **Export**.
Parameters	Name / Type / Description <SetupName> / String / Name of the setup. <filePath> / String / Full path for file export.
Return Value	None

Python Syntax	ExportParametricSetupTable (<SetupName>, <filePath>)
Python Example	`oModule.ExportParametricSetupTable('ParametricSetup1', 'E:/Files/ParametricSetup1_Table.csv')`

ExportRespSurfaceMinMaxTable

Exports min-max table from a response surface to a file

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportRespSurfaceMinMaxTable(<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceMinMaxTable("DesignOfExperimentsSetup1",` `"C:/temp/DesignOfExperimentsSetup1_Min-Max_Search.csv")`

ExportRespSurfaceRefinePoints

Exports refinement points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Refinement Points** option under **View**, Click on **Export...**.
Parameters	Name | Type | Description <*DOEName*> | String | Name of the Design of Experiments (DOE) setup. <*FileName*> | String | Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceRefinePoints(<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceRefinePoints("DesignOfExperimentsSetup1",` `"C:/temp/DesignOfExperimentsSetup1_Refine_Points.csv")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportRespSurfaceResponsePoints

Exports response points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Response Points** option under **View**, Click on **Export...**.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceResponsePoints (*<DOEName>*, *<FileName>*)
Python Example	`oModule.ExportRespSurfaceResponsePoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportRespSurfaceVerificationPoints

Exports verification points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Verification Points** option under **View**, Click on **Export...**.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportRespSurfaceVerificationPoints (<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceVerificationPoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Veri_Points.csv")`

GenerateVariationData [Parametric]

Generates variation data before parametric solve for CAD integrated project.

UI Access	Right click on the parametric setup in the project tree and select **Generate Variation Data**		
Parameters	Name	Type	Description
	<*SetupName*>	String	Name of the setup to use for the variation data generation
Return Value	None		

Python Syntax	GenerateVariationData <*SetupName*>
Python Example	`oModule.GenerateVariationData("ParametricSetup1")`

GetChildNames [Optimetrics]

If used without a specific optimization setup name, gets a list of all setups for all types. If a with a specific setup name, returns names for that optimization setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	Name: typeName; Type: text string; Description: Optional, default to get all types of setup names. Or one of type name return in GetChildTypes(). Also, the type name can be used without the prefix “Opti”.
Return Value	Array of setup names.

Python Syntax	GetChildNames()

Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")`
	`arrAllSetup = oOptimModule.GetChildNames()`
	`arrParmSetup = oOptimModule.GetChildNames("'OptiParametric'")`
	`arrOptimizeSetup = oOptimModule.GetChildNames("'Optimization'")`

GetChildObject [Optimetrics]

Gets a Setup Object of the Optimetrics module

UI Access	NA
Parameters	Name: Setup Name; Type: text string; Description: A optimetrics setup name, names returned by the GetChildNames().
Return Value	A script object for the setup See discussion of Optimetrics Setup Objects in Object Script Property Function Summary.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetChildObject()
Python Example	`oParamSetup = oOptModule.GetChildObject('ParametricSetup1')` `oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')`

GetChildTypes [Optimetrics]

Use: Gets child types of queried Optimetrics module.

Syntax: GetChildTypes()

Return Value: Array of text string, it can be an empty array if there is no setup is defined. There are six types of setup, they are ['OptiParametric', 'OptiOptimization', 'OptiSensitivity', 'OptiStatistical', 'OptiDesignExplorer', 'OptiDXDOE'].

Python Syntax	GetChildTypes ()
Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")` `arrSetupTypes = oOptimModule.GetChildTypes()`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetObjPath [Design]

Obtains the path to the design.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the design.

Python Syntax	GetObjPath()
Python Example	`oDesign.GetObjPath()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetOptimetricResult

Returns an Optimetric calculation. The specific calculation is determined by the setup.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SetupName*></td><td>String</td><td>Optimetrics setup name.</td></tr><tr><td><*vars*></td><td>Array</td><td>Array containing string variable names. Use the **Sweep Definitions** tab in the UI or the <SweepDefs> parameter in the InsertSetup script to determine appropriate inputs.</td></tr><tr><td><*values*></td><td>Array</td><td>*Optional*. Array containing string values. When multiple variables and values are provided, the order must be the same in both the <vars> and <values> arrays. The first variable is paired with the first value, the second variable is paired with the second value, and so on.</td></tr></table>
Return Value	Calculation result. If the setup contains more than one calculation, the output will be an array of values.

Python Syntax	GetOptimetricResult(<*SetupName*>, <*vars*>, <*values*>)
Python Example	`oModule.GetOptimetricResult('ParametricSetup1',['AR','Re'],['4.64','6e+04'])`

GetOptimetricsResult

Get an existing Optimization, Sensitivity, Statistical, Parametric or DesignXplorer result.

UI Access	Right click on the desired Optimetrics setup in the project tree and choose **View Analysis Result…** On the **Post**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Analysis Display dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Optimization, Sensitivity, Statistical, or DesignXplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat, or txt..</td></tr><tr><td>[useFullOutputName]</td><td>Boolean</td><td>Optional: defaulted to false. If set to true values will be printed with units. This parameter is ignored for Optimization and Statistical results.</td></tr></table>
Return Value	None

Python Syntax	GetOptimetricsResult (<SetupName>, <FileName>, [useFullOutputName])
Python Example	`oModule.GetOptimetricsResult (` `"StatisticalSetup1", "c:/exportdir/test.csv",false)`

GetPropNames [Optimetrics]

Use: Always returns the empty set for Optimetrics objects since they do not have properties.

Syntax: GetPropNames(bIncludeReadOnly)

Return Value: Returns empty set.

Parameters: `bIncludeReadOnly—optional, default to True.`

Python Syntax	GetPropNames ()
Python Example	`oOptModule.GetPropNames()` `oOptModule.GetPropNames(True)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oOptModule.GetPropNames(False)`

GetPropValue [Optimetrics]

Returns the property value for a setup property.

UI Access	NA
Parameters	Name | Type | Description property-path | | a child object's property path. See property path discussion here.
Return Value	Returns the value of an setup property.

Python Syntax	GetPropValue(propPath)
Python Example	`oOptModule.GetPropValue("OptimizationSetup1\Optimizer") //get the optimizer name for OptimizationSetup1` `oOptModule.GetPropValue("OptimizationSetup1\Optimizer\Choices") //Get the menu property's menu items. In this case all Optimizer names.`

GetSetupNames [Optimetrics]

Gets a list of Optimetrics setup names

UI Access	NA
Parameters	Name | Type | Description None | |

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	IAnsoftCollectionObj – a collection of Optimetrics setup names

Python Syntax	GetSetupNames()
Python Example	`oModule = oDesign.GetModule("Optimetrics")` `setupNames = oModule.GetSetupNames()`

GetSetupNamesByType [Optimetrics]

Gets a list of Optimetrics setup names by type.

UI Access	NA
Parameters	Name | Type | Description <Optimetrics type> | String | Examples: parametric, optimization, statistical, sensitivity
Return Value	Array of Optimetrics setup names of the given type.

Python Syntax	GetSetupNamesByType (<Optimetrics type>)
Python Example	`for name in oModule.GetSetupNamesByType("optimization")` `  AddInfoMessage(str(name))`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportSetup

Import an Optimetric setup from a file.

<table>
<tr><td>UI Access</td><td>NA</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SetupTypeName></td><td>String</td><td>Must be one of "OptiParametric" , "OptiOptimization", "OptiSensitivity", "OptiStatistical", or "OptiDesignExplorer".</td></tr>
<tr><td><SetupInfo></td><td>Array</td><td>[NAME:<SetupName>", "FilePath"]

<SetupName>

Type: <string>; name of the setup.

<FilePath>

Type : <string: file path>; must be a valid file path and name.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	ImportSetup (<SetupTypeName>, <SetupInfo>)
Python Example	oModule.ImportSetup ("OptiStatistical", ["NAME:StatisticalSetup1", "c:/importdir/mySetupInfoFile"])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PasteSetup [Optimetrics]

Pastes the specified Optimetrics setup.

UI Access	NA
Parameters	Name | Type | Description <SetupName> | String | Name of the Setup
Return Value	None

Python Syntax	PasteSetup (<SetupName>)
Python Example	`oModule.PasteSetup ("OptimizationSetup1")`

RenameSetup [Optimetrics]

Renames the specified Optimetrics setup.

UI Access	**Right-click the setup in the project tree, and then click Rename** on the shortcut menu.
Parameters	Name | Type | Description <OldName> | String | The name that needs to be replaced <NewName> | String | Replacement name
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	RenameSetup (<OldName> <NewName>)
Python Example	`oModule.RenameSetup ("OptimizationSetup1" "MyOptimization")`

SetPropValue [Optimetrics]

Sets the property value for the active Optimetrics setup.

UI Access	Set Property value on Optimetrics objects
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>Property path</td><td>text string</td><td>Setup property path. See discussion of Property Path</td></tr><tr><td>new Value</td><td>Text String, Number, or Boolean</td><td>New value data type is depending on the property type,</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Python Syntax	SetPropValue(propPath, newValue)
Python Example	`oOptModule.SetPropValue("ParametricSetup1\Enabled", False) //disable ParametricSetup1` `oOptModule.SetPropValue("OptimizationSetup1/Optimizer","Quasi Newton")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SolveAllSetup

Solves all Optimetrics setups.

UI Access	Right-click on **Optimetrics** in Project Manager and select **Analyze>All** from context menu
Parameters	Name / Type / Description *<isBlocking>* / Boolean / An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	SolveAllSetup()
Python Example	`oModule.SolveAllSetup()`

SolveSetup [Optimetrics]

Solves the specified Optimetrics setup.

UI Access	Right-click the setup in the project tree, and then click **Analyze** on the shortcut menu.
Parameters	Name / Type / Description *<SetupName>* / String / Name of the setup to be solved *<isBlocking>* / Boolean / An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` com-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	mand.
Return Value	None

Python Syntax	SolveSetup (<*SetupName*>)
Python Example	`oModule.SolveSetup ("OptimizationSetup1")`

ValidateSetup [Parametric]

Performs a validation of all design variations in a parametric setup.

For information on reviewing the output produced by this script command, see: Validating a Parametric Setup

UI Access	Under *Optimetrics* in the Project Manager, right-click the ***<Parametric Setup Name>*** and then click **Validate Parametric Setup** on the shortcut menu.
Parameters	Name | Type | Description <SetupName> | String | Name of the parametric setup.
Return Value	None

Python Syntax	ValidateSetup (<SetupName>)
Python Example	`oModule.ValidateSetup('ParametricSetup1')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

General Commands Recognized by the Optimetrics Module

Following are general script commands recognized by the **Optimetrics** module:

CopySetup

DeleteSetups [Optimetrics]

DistrbutedAnalyzeSetup

EditSetup

EnableSetup

ExportDXConfigFile

ExportOptimetricsProfile

ExportOptimetricsResult

ExportParametricResults

ExportRespSurfaceMinMaxTable

ExportRespSurfaceRefinePoints

ExportRespSurfaceResponsePoints

ExportRespSurfaceVerificationPoints

ExportDOEResponseCurve

ExportDOEResponseCurveSlices

ExportDOEResponseSurface

ExportDOELocalSensitivity

ExportDOELocalSensitivityCurve

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChildNames [Optimetrics]

GetChildObject [Optimetrics]

GetChildTypes [Optimetrics]

GetName

GetObjPath [Editor]

GetOptimetricResult

GetPropNames [Optimetrics]

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

ImportSetup

InsertSetup

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

SolveSetup [Optimetrics]

SolveAllSetup

CopySetup

Copy the specified Optimetrics setup.

UI Access	NA

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of the setup.</td></tr></table>
Return Value	None.

Python Syntax	CopySetup (<*SetupName*>)
Python Example	`oModule.CopySetup ("OptimizationSetup1")`

DeleteSetups [Optimetrics]

Deletes the specified Optimetrics setups.

UI Access	Right-click the setup in the project tree, and then click **Delete** on the shortcut menu
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><NameArray></td><td>Array of Strings</td><td>An Array of Setup Names</td></tr></table>
Return Value	None

Python Syntax	DeleteSetups (<NameArray>)
Python Example	`oModule.DeleteSetups (["OptimizationSetup1"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DistributedAnalyzeSetup

Distributes all variable value instances within a parametric sweep to different machines already specified from within the user interface

UI Access	Right-click the parametric setup name in the project tree and select Distribute Analysis.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ParametricSetupName*></td><td>String</td><td>Name of the Setup</td></tr><tr><td><*isBlocking*></td><td>Boolean</td><td>An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.</td></tr></table>
Return Value	None

Python Syntax	DistributedAnalyzeSetup (<*ParametricSetupName*>)
Python Example	`oModule.DistributedAnalyzeSetup("ParametricSetup1")`

EnableSetup

Enables and disables a defined optimetrics analysis setup.

UI Access	Right-click on a setup in the project tree, select **Enable Setup** or **Disable Setup**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of specified setup.</td></tr><tr><td><*Enable*></td><td>Boolean</td><td>Determines whether enable or disable a setup.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• **True** - enable setup. • **False** - disable setup.
Return Value	None.		

Python Syntax	EnableSetup(<*SetupName*>, <*Enable*>)
Python Example	`oModule.EnableSetup("OptimizationSetup1", True)`

ExportDXConfigFile

Create an xml file with the setup information for Design Xplorer

UI Access	Right click on the Design Xplorer setup in the project tree and choose **Export External Connector Addin Configuration…**		
Parameters	Name	Type	Description
	<SetupName>	String	Must be one of existing DesignExplorer setup names
	<FileName>	String	Must be a valid file path and name
Return Value	None		

Python Syntax	ExportDXConfigFile (<SetupName>, <FileName>)
Python Example	`oModule.ExportDXConfigFile ("DesignXplorerSetup1",` `"c:/exportdir/DXSetup1.xml")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOptimetricsProfile

Export Optimetrics profile data

UI Access	Right click on the Optimetrics setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Profile** tab and click on the **Export** button.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Parametric, Optimization, Sensitivity, Statistical or DesignXplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat, or txt</td></tr><tr><td>[profileNum]</td><td>String</td><td>Must be a numeric string. Optional: defaulted to last profile number. It should be a zero indexed profile number.</td></tr></table>
Return Value	None

Python Syntax	ExportOptimetricsProfile (<SetupName>, <FileName>, [profileNum]
Python Example	`oModule.ExportOptimetricsProfile` `("StatisticalSetup1", "c:/exportdir/test.csv")`

ExportOptimetricsResult

Export an existing Optimization, Sensitivity, Statistical or DesignXplorer result. (Does not export Parametric results.)

UI Access	Right click on the desired Optimetrics setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Optimization, Sensitivity, Statistical, or DesignXplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat, or txt..</td></tr><tr><td>[useFullOutputName]</td><td>Boolean</td><td>Optional: defaulted to false. If set to true values will be printed with units. This parameter is ignored for Optimization and Statistical results.</td></tr></table>
Return Value	None

Python Syntax	ExportOptimetricsResult (<SetupName>, <FileName>, [useFullOutputName])
Python Example	`oModule.ExportOptimetricsResult (` `"StatisticalSetup1", "c:/exportdir/test.csv",false)`

ExportParametricResults

Export existing Parametric results.

UI Access	Right click on the desired Parametric setup in the project tree and choose **View Analysis Result…** On the **Post Analysis Display** dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><SetupName></td><td>String</td><td>Must be one of the existing Parametric setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name with extension of csv, tab, dat or txt</td></tr><tr><td><bOutputUnits></td><td>Boolean</td><td>If set to true, values will be printed with units</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportParametricResults (<SetupName>, <FileName>, <bOutputUnits>)
Python Example	`oModule.ExportParametricResults(` `"ParametricSetup1", "c:/exportdir/test.csv", False)`

ExportParametricSetupTable

Exports the parametric setup table as a CSV file.

UI Access	Double-click parametric setup. Select **Table** tab. Click **Export**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><filePath></td><td>String</td><td>Full path for file export.</td></tr></table>
Return Value	None

Python Syntax	ExportParametricSetupTable (<SetupName>, <filePath>)
Python Example	`oModule.ExportParametricSetupTable('ParametricSetup1', 'E:/Files/Parametric-Setup1_Table.csv')`

ExportRespSurfaceMinMaxTable

Exports min-max table from a response surface to a file

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceMinMaxTable(*<DOEName>*, *<FileName>*)
Python Example	`oModule.ExportRespSurfaceMinMaxTable("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Min-Max_Search.csv")`

ExportRespSurfaceRefinePoints

Exports refinement points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Refinement Points** option under **View**, Click on **Export...**.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceRefinePoints(*<DOEName>*, *<FileName>*)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.ExportRespSurfaceRefinePoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Refine_Points.csv")`

ExportRespSurfaceResponsePoints

Exports response points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Response Points** option under **View**, Click on **Export...**.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceResponsePoints (<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceResponsePoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportRespSurfaceVerificationPoints

Exports verification points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Veri-**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	fication Points option under **View**, Click on **Export...**.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceVerificationPoints (<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceVerificationPoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Veri_Points.csv")`

ExportDOEResponseCurve

Exports response curve from a response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.ExportDOEResponseCurve("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOEResponseCurveSlices

Exports response curve slices from a response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (*<FileName>*)
Python Example	`oModule.ExportDOEResponseCurveSlices("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOEResponseSurface

Exports response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dia-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	log.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOEResponseSurface("DesignOfExperimentsSetup1", "C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOELocalSensitivity

Exports local sensitivity to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.ExportDOELocalSensitivity("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOELocalSensitivityCurve

Exports local sensitivity curves to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup. *<FileName>* / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (*<FileName>*)
Python Example	`oModule.ExportDOELocalSensitivityCurve("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

GenerateVariationData [Parametric]

Generates variation data before parametric solve for CAD integrated project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right click on the parametric setup in the project tree and select **Generate Variation Data**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of the setup to use for the variation data generation</td></tr></table>
Return Value	None

Python Syntax	GenerateVariationData <*SetupName*>
Python Example	`oModule.GenerateVariationData("ParametricSetup1")`

GetChildNames [Optimetrics]

If used without a specific optimization setup name, gets a list of all setups for all types. If a with a specific setup name, returns names for that optimization setup.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>typeName</td><td>text string</td><td>Optional, default to get all types of setup names. Or one of type name return in GetChildTypes(). Also, the type name can be used without the prefix “Opti”.</td></tr></table>
Return Value	Array of setup names.

Python Syntax	GetChildNames()

Python Example	`oOptimModule = oDesign.GetChildObject(“Optimetrics”)`
	`arrAllSetup = oOptimModule.GetChildNames()`
	`arrParmSetup = oOptimModule.GetChildNames(“'OptiParametric'”)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`arrOptimizeSetup = oOptimModule.GetChildNames("'Optimization'")`

GetChildObject [Optimetrics]

Gets a Setup Object of the Optimetrics module

UI Access	NA
Parameters	Name | Type | Description Setup Name | text string | A optimetrics setup name, names returned by the GetChildNames().
Return Value	A script object for the setup See discussion of Optimetrics Setup Objects in Object Script Property Function Summary.

Python Syntax	GetChildObject()
Python Example	`oParamSetup = oOptModule.GetChildObject('ParametricSetup1')` `oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')`

GetChildTypes [Optimetrics]

Use: Gets child types of queried Optimetrics module.

Syntax: GetChildTypes()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: Array of text string, it can be an empty array if there is no setup is defined. There are six types of setup, they are ['OptiParametric', 'OptiOptimization', 'OptiSensitivity', 'OptiStatistical', 'OptiDesignExplorer', 'OptiDXDOE'].

Python Syntax	GetChildTypes ()
Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")` `arrSetupTypes = oOptimModule.GetChildTypes()`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetObjPath [Design]

Obtains the path to the design.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the design.

Python Syntax	GetObjPath()
Python Example	`oDesign.GetObjPath()`

GetOptimetricResult

Returns an Optimetric calculation. The specific calculation is determined by the setup.

UI Access	N/A
Parameters	

Name	Type	Description
<*SetupName*>	String	Optimetrics setup name.
<*vars*>	Array	Array containing string variable names. Use the **Sweep Definitions** tab in the UI or the <SweepDefs> parameter in the InsertSetup script to determine appropriate inputs.
<*values*>	Array	*Optional*. Array containing string values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			When multiple variables and values are provided, the order must be the same in both the <vars> and <values> arrays. The first variable is paired with the first value, the second variable is paired with the second value, and so on.
Return Value	Calculation result. If the setup contains more than one calculation, the output will be an array of values.		

Python Syntax	GetOptimetricResult(<*SetupName*>, <*vars*>, <*values*>)
Python Example	`oModule.GetOptimetricResult('ParametricSetup1',['AR','Re'],['4.64','6e+04'])`

GetPropNames [Optimetrics]

Use: Always returns the empty set for Optimetrics objects since they do not have properties.

Syntax: GetPropNames(bIncludeReadOnly)

Return Value: Returns empty set.

Parameters: `bIncludeReadOnly—optional, default to True.`

Python Syntax	GetPropNames ()

Python Example	`oOptModule.GetPropNames()`
	`oOptModule.GetPropNames(True)`
	`oOptModule.GetPropNames(False)`

GetPropValue [Optimetrics]

Returns the property value for a setup property.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	Name | Type | Description property-path | | a child object's property path. See property path discussion here.
Return Value	Returns the value of an setup property.

Python Syntax	GetPropValue(propPath)
Python Example	`oOptModule.GetPropValue("OptimizationSetup1\Optimizer") //get the optimizer name for OptimizationSetup1` `oOptModule.GetPropValue("OptimizationSetup1\Optimizer\Choices") //Get the menu property's menu items. In this case all Optimizer names.`

GetSetupNames [Optimetrics]

Gets a list of Optimetrics setup names

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	IAnsoftCollectionObj – a collection of Optimetrics setup names

Python Syntax	GetSetupNames()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule = oDesign.GetModule("Optimetrics")` `setupNames = oModule.GetSetupNames()`

GetSetupNamesByType [Optimetrics]

Gets a list of Optimetrics setup names by type.

UI Access	NA
Parameters	Name | Type | Description <Optimetrics type> | String | Examples: parametric, optimization, statistical, sensitivity
Return Value	Array of Optimetrics setup names of the given type.

Python Syntax	GetSetupNamesByType (<Optimetrics type>)
Python Example	`for name in oModule.GetSetupNamesByType("optimization")` `  AddInfoMessage(str(name))`

ImportSetup

Import an Optimetric setup from a file.

UI Access	NA
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<SetupTypeName>	String	Must be one of "OptiParametric" , "OptiOptimization", "OptiSensitivity", "OptiStatistical", or "OptiDesignExplorer".

	<SetupInfo>	Array	`[NAME:<SetupName>", "FilePath"]`
			`<SetupName>`
			Type: <string>; name of the setup.
			`<FilePath>`
			Type : <string: file path>; must be a valid file path and name.

Return Value	None		

Python Syntax	ImportSetup (<SetupTypeName>, <SetupInfo>)

Python Example	`oModule.ImportSetup ("OptiStatistical",`
	`["NAME:StatisticalSetup1",`
	`"c:/importdir/mySetupInfoFile"])`

PasteSetup [Optimetrics]

Pastes the specified Optimetrics setup.

UI Access	NA		
Parameters	Name	Type	Description
	<SetupName>	String	Name of the Setup

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	PasteSetup (<SetupName>)
Python Example	`oModule.PasteSetup ("OptimizationSetup1")`

RenameSetup [Optimetrics]

Renames the specified Optimetrics setup.

UI Access	**Right-click the setup in the project tree, and then click Rename** on the shortcut menu.
Parameters	Name / Type / Description <OldName> / String / The name that needs to be replaced <NewName> / String / Replacement name
Return Value	None

Python Syntax	RenameSetup (<OldName> <NewName>)
Python Example	`oModule.RenameSetup ("OptimizationSetup1" "MyOptimization")`

SetPropValue [Optimetrics]

Sets the property value for the active Optimetrics setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Set Property value on Optimetrics objects
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>Property path</td><td>text string</td><td>Setup property path. See discussion of Property Path</td></tr><tr><td>new Value</td><td>Text String, Number, or Boolean</td><td>New value data type is depending on the property type,</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Python Syntax	SetPropValue(propPath, newValue)
Python Example	`oOptModule.SetPropValue("ParametricSetup1\Enabled", False) //disable ParametricSetup1` `oOptModule.SetPropValue("OptimizationSetup1/Optimizer","Quasi Newton")`

SolveAllSetup

Solves all Optimetrics setups.

UI Access	Right-click on **Optimetrics** in Project Manager and select **Analyze>All** from context menu
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*isBlocking*></td><td>Boolean</td><td>An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	SolveAllSetup()
Python Example	`oModule.SolveAllSetup()`

SolveSetup [Optimetrics]

Solves the specified Optimetrics setup.

UI Access	Right-click the setup in the project tree, and then click **Analyze** on the shortcut menu.
Parameters	Name / Type / Description <*SetupName*> / String / Name of the setup to be solved <*isBlocking*> / Boolean / An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	SolveSetup (<*SetupName*>)
Python Example	`oModule.SolveSetup ("OptimizationSetup1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ValidateSetup [Parametric]

Performs a validation of all design variations in a parametric setup.

For information on reviewing the output produced by this script command, see: Validating a Parametric Setup

UI Access	Under *Optimetrics* in the Project Manager, right-click the ***<Parametric Setup Name>*** and then click **Validate Parametric Setup** on the shortcut menu.
Parameters	Name / Type / Description <SetupName> / String / Name of the parametric setup.
Return Value	None

Python Syntax	ValidateSetup (<SetupName>)
Python Example	`oModule.ValidateSetup('ParametricSetup1')`

Parametric Script Commands

EditSetup [Parametric]

ExportParametricSetupTable

GenerateVariationData [Parametric]

InsertSetup [Parametric]

ValidateSetup [Parametric]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditSetup [Parametric]

Modifies an existing parametric setup

UI Access	Right-click the setup in the project tree, and then click **Properties** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr><tr><td><ParametricParams></td><td>List</td><td>List that defines the parameters of the parametric setup; examples are listed below.</td></tr></table>
Return Value	None

Python Syntax	EditSetup (<SetupName>, <ParametricParams>)
Python Example	See EditSetup [Optimization]

ExportParametricSetupTable

Exports the parametric setup table as a CSV file.

UI Access	Double-click parametric setup. Select **Table** tab. Click **Export**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><filePath></td><td>String</td><td>Full path for file export.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportParametricSetupTable (<SetupName>, <filePath>)
Python Example	`oModule.ExportParametricSetupTable('ParametricSetup1', 'E:/Files/ParametricSetup1_Table.csv')`

GenerateVariationData [Parametric]

Generates variation data before parametric solve for CAD integrated project.

UI Access	Right click on the parametric setup in the project tree and select **Generate Variation Data**
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup to use for the variation data generation
Return Value	None

Python Syntax	GenerateVariationData <*SetupName*>
Python Example	`oModule.GenerateVariationData("ParametricSetup1")`

InsertSetup [Parametric]

Inserts a new parametric setup.

UI Access	Right-click the **Optimetrics** folder in the project tree, and then click **Add> Parametric** on the shortcut menu.
Parameters	Name | Type | Description <Parametric Params> | Array | `["NAME:<SetupName>", "SaveFields:=",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<SaveField>, <StartingPoint>, "Sim. Setups:=", <SimSetups>, <SweepDefs>, <SweepOps>, ["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>), ... ["NAME:Goal", <OptiGoalSpec>]]
	<SetupName>	String	Name of the parametric setup.
	<SimSetups>	Array of Strings	An array of Twin Builder solution setup names.
	<SweepDefs>	Array	["NAME:Sweeps", ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>), ... ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>]]
	<SweepData>	String	"<SweepType>, <StartV>, <StopV>, <StepV>"
	<SweepType>	String	The type of sweep data.
	<SyncNum>	Integer	SweepDatas with the same value are synchronized.
	<SweepOps>		["NAME:Sweep Operations", "<OpType>:=, [<VarValue>, …, <VarValue>], ... <OpType>:=, [<VarValue>, …, <VarValue>]]
	<OpType>	String	The sweep operation type.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	InsertSetup ("OptiParametric", <ParametricParams>)
Python Example	oModule.InsertSetup("OptiParametric", ["NAME:ParametricSetup1", "SaveFields:=", true, ["NAME:StartingPoint"], "Sim. Setups:=", ["Setup1"], ["NAME:Sweeps", ["NAME:SweepDefinition", "Variable:=", "$width", "Data:=", "LIN 12mm 17mm 2.5mm", "OffsetF1:=", false, "Synchronize:=", 0], ["NAME:SweepDefinition", "Variable:=", "$length", "Data:=", "LIN 8mm 12mm 2mm", "OffsetF1:=", false, "Synchronize:=", 0]],

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  [
    "NAME:Sweep Operations"
  ],
  [
    "NAME:Goals",
    [
      "NAME:Goal",
      "Solution:=", "Setup1 : LastAdaptive",
      "Calculation:=", "returnloss",
      "Context:=", ""
      [
        "NAME:Ranges",
        "Range:=",
        [
        "Var:=", "Freq", "Type:=","s"
        "Start:=", "8GHz",
        "Stop:=", "8GHz"
        ]
      ],
      [
        "NAME:Goal",
        "Solution:=", "Setup1 : LastAdaptive"
        "Calculation:=", "reflect",
        "Context:=", "",
        [
          "NAME:Ranges",
          "Range:=",
          [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "Var:=", "Freq",
                "Type:=","s",
                "Start:=", "8GHz",
                "Stop:=", "8GHz"
            ]
        ]
    ]
  ]
])
```

ValidateSetup [Parametric]

Performs a validation of all design variations in a parametric setup.

For information on reviewing the output produced by this script command, see: Validating a Parametric Setup

UI Access	Under *Optimetrics* in the Project Manager, right-click the ***<Parametric Setup Name>*** and then click **Validate Parametric Setup** on the shortcut menu.
Parameters	Name | Type | Description <SetupName> | String | Name of the parametric setup.
Return Value	None

Python Syntax	ValidateSetup (<SetupName>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.ValidateSetup('ParametricSetup1')`

Optimization Script Commands

EditSetup [Optimization]

InsertSetup [Optimization]

EditSetup [Optimization]

Modifies an existing optimization setup.

UI Access	Right-click the setup in the Project Manager tree, and then click **Properties** from the shortcut menu
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr><tr><td><OptimizationParams></td><td>List</td><td>Parameters that define the setup; examples are listed below.</td></tr></table>
Return Value	None

Python Syntax	EditSetup (<SetupName>, <OptimizationParams>)

Python Example	`oModule.EditSetup("OptimizationSetup1",`
	`[`
	`  "NAME:OptimizationSetup1",`
	`  "UseFastCalculationUpdateAlgo:=", False,`
	`  "FastCalcOptCtrledByUser:=", False,`
	`  "IsEnabled:=", True,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"SaveSolutions:=", False,
[
  "NAME:StartingPoint"
],
"Optimizer:=", "Quasi Newton",
[
  "NAME:AnalysisStopOptions",
  "StopForNumIteration:=" , True,
  "StopForElapsTime:=", False,
  "StopForSlowImprovement:=", False,
  "StopForGrdTolerance:=", False,
  "MaxNumIteration:=",1000,
  "MaxSolTimeInSec:=",3600,
  "RelGradientTolerance:=", 0,
  "MinNumIteration:=", 10
],
"CostFuncNormType:=", "L2",
"PriorPSetup:=", "",
"PreSolvePSetup:=", True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  [
    "NAME:Variables"
  ],
  [
    "NAME:LCS"
  ],
  [
    "NAME:Goals",
    [
      "NAME:Goal",
      "ReportType:=", "Standard",
      "Solution:=", "TR",
      [
        "NAME:SimValueContext",
        "SimValueContext:=", [1,0,2,0,False,False,-1,1,0,1,1,"",0,0
      ]
    ],
    "Calculation:=", "acosh(Time)",
    "Name:=", "Time",
    [
      "NAME:Ranges",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Range:=", ["Var:=", "Time","Type:=", "a"]
      ],
      "Condition:=", "==",
      [
        "NAME:GoalValue",
        "GoalValueType:=", "Independent",
        "Format:=", "Real/Imag",
        "bG:=", ["v:=", "[1;]"]
      ],
      "Weight:=", "[1;]"
    ],
    "Acceptable_Cost:="    , 0,
    "Noise:=", 0.0001,
    "UpdateDesign:=", False,
    "UpdateIteration:=", 5,
    "KeepReportAxis:=", True,
    "UpdateDesignWhenDone:=", True
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

InsertSetup [Optimization]

Inserts a new optimization setup.

<table>
<tr><td>UI Access</td><td>Right-click the Optimetrics folder in the project tree, and then click Add > Optimization on the shortcut menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OptimizationParams></td><td>Array</td><td>["NAME:<SetupName>", "SaveFields:=",
<SaveField>, <StartingPoint>, "Optimizer:=",
<Optimizer>,
"MaxIterations:=", <MaxIter>, "PriorPSetup:=",
<PriorSetup>, "PreSolvePSetup:=", <Preceed>,
<OptimizationVars>, <Constraint>,
["NAME:Goals",
["NAME:Goal",
<OptiGoalSpec>, <OptimizationGoalSpec>], ...
["NAME:Goal", <OptiGoalSpec>,
<OptimizationGoalSpec>]],
"Acceptable_Cost:=", <AcceptableCost>, "Noise:=",
<Noise>, "UpdateDesignWhenDone:=", <UpdateDesign></td></tr>
<tr><td><OptimizationVars></td><td>Array</td><td>["NAME:Variables", "VarName:=",
["i:=",<IncludeVar>, "Min:=", <MinV>, "Max:=",
<MaxV>,
"MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>],
……. "VarName:=", ["i:=", <IncludeVar>,
"Min:=", <MinV>, "Max:=", <MaxV>,
"MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>]]</td></tr>
<tr><td><MinStepV></td><td>VarValue</td><td>The minimum step of the variable.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<MaxStepV>	VarValue	The maximum step of the variable.
	<AcceptableCost>	Double	The acceptable cost value for the optimizer to stop.
	<Noise>	Double	The noise of the design.
	<UpdateDesign>	Boolean	Specifies whether or not to apply the optimal variation to the design after the optimization is done.
	<OptimizationGoalSpec>	Array	"Condition:=", <OptimizationCond>, ["NAME:GoalValue", "GoalValeType:=", <GoalValueType>, "Format:=", <GoalValueFormat>, "bG:=", ["v:=", <GoalValue>]], "Weight:=", <Weight>]
	<OptimizationCond>	String	Either "<=", "==", or ">="
	<GoalValueType>	String	Either "Independent" or "Dependent"
	<GoalValueFormat>	String	Either "Real/Imag" or "Mag/Ang".
	<GoalValue>	String	Value in string. Value can be a real number, complex number, or expression.
Return Value	None		

<MinStepV>

Type : <VarValue>

The minimum step of the variable.

<MaxStepV>

Type: <VarValue>

The maximum step of the variable.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<AcceptableCost>

Type: <double>

The acceptable cost value for the optimizer to stop.

<Noise>

Type: <double>

The noise of the design.

<UpdateDesign>

Type: <bool>

Specifies whether or not to apply the optimal variation to the design after the optimization is done.

```
<OptimizationGoalSpec>
   "Condition:=", <OptimizationCond>,
  Array("NAME:GoalValue", "GoalValueType:=",
  <GoalValueType>,
   "Format:=", <GoalValueFormat>, "bG:=",
  Array("v:=", <GoalValue>)), "Weight:=", <Weight>)
```

<OptimizationCond>

Type: <string>

Either "<=", "==", or ">="

<GoalValueType>

Type: <string>

Either "Independent" or "Dependent"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<GoalValueFormat>

Type:<string>

Either "Real/Imag" or "Mag/Ang".

<GoalValue>

Type: <string>

Value in string. Value can be a real number, complex number, or expression.

Python Syntax	InsertSetup ("OptiOptimization", <OptimizationParams>)
Python Example	oModule.InsertSetup("OptiOptimization", ["NAME:OptimizationSetup1", "SaveFields:=", false, _ ["NAME:StartingPoint", "$length:=", "8mm", "$width:=", "14.5mm"], "Optimizer:=", "Quasi Newton", "MaxIterations:=", 100, "PriorPSetup:=", "ParametricSetup1", "PreSolvePSetup:=", true, ["NAME:Variables", "$length:=", ["i:=", true, "Min:=", "6mm", "Max:=", "18mm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"MinStep:=", "0.001mm", "MaxStep:=",
"1.2mm"],
"$width:=", ["i:=", true, "Min:=",
"6.5mm", "Max:=", "19.5mm",
"MinStep:=", "0.001mm", "MaxStep:=",
"1.3mm"]],
["NAME:LCS"],
["NAME:Goals",
["NAME:Goal",
"Solution:=", "Setup1 : LastAdaptive",
"Calculation:=", "reflect",
"Context:=", "",
["NAME:Ranges",
"Range:=", ["Var:=", "Freq",
"Type:=", "s",
"Start:=", "8GHz", "Stop:=", "8GHz"]],
"Condition:=", "&lt;=",
["NAME:GoalValue",
"GoalValueType:=", "Independent",
"Format:=", "Real/Imag",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"bG:=", ["v:=", "[0.0001]"]], "Weight:=", "[1]"]], "Acceptable_Cost:=", 0.0002, "Noise:=", 0.0001, "UpdateDesign:=", true, "UpdateIteration:=", 5, "KeepReportAxis:=", true, "UpdateDesignWhenDone:=", true])

Sensitivity Script Commands

EditSetup [Sensitivity]

InsertSetup [Sensitivity]

EditSetup [Sensitivity]

Modifies an existing sensitivity setup.

UI Access	Right-click the setup in the project tree, and then click Properties on the shortcut menu
Parameters	Name | Type | Description <SetupName> | String | Name of the Setup | |

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	EditSetup (<SetupName>, <SensitivityParams>)
Python Example	oModule.EditSetup("OptimizationSetup1", ["NAME:OptimizationSetup1", "UseFastCalculationUpdateAlgo:=", False, "FastCalcOptCtrledByUser:=", False, "IsEnabled:=", True, "SaveSolutions:=", False, ["NAME:StartingPoint"], "Optimizer:=", "Quasi Newton", ["NAME:AnalysisStopOptions", "StopForNumIteration:=", True, "StopForElapsTime:=", False, "StopForSlowImprovement:=", False, "StopForGrdTolerance:=" , False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"MaxNumIteration:=", 1001,
"MaxSolTimeInSec:=", 3600,
"RelGradientTolerance:=", 0,
"MinNumIteration:=", 10
],
"CostFuncNormType:=", "L2",
"PriorPSetup:=", "",
"PreSolvePSetup:=", True,
[
"NAME:Variables"
],
[
"NAME:LCS"
],
[
"NAME:Goals",
[
"NAME:Goal",
"ReportType:=", "Standard",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Solution:=", "TR3",
[
"NAME:SimValueContext",
"SimValueContext:=", [1,0,2,0,False,False,-1,1,0,1,1,"",0,0]
],
"Calculation:=", "mag(DIFF1.VAL)",
"Name:=", "DIFF1.VAL",
[
"NAME:Ranges",
"Range:=", [
"Var:=", "Time","Type:=", "a"]
],
"Condition:=", "==",
[
"NAME:GoalValue",
"GoalValueType:=", "Independent",
"Format:=", "Real/Imag",
"bG:=", ["v:=", "[1;]"]
],
"Weight:=", "[1;]"
]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
],
"Acceptable_Cost:=", 0,
"Noise:=", 0.0001,
"UpdateDesign:=", False,
"UpdateIteration:=", 5,
"KeepReportAxis:=", True,
"UpdateDesignWhenDone:=", True
])
```

Statistical Script Commands

EditSetup [Statistical]

InsertSetup Statistical

EditSetup [Statistical]

Modifies an existing statistical setup.

UI Access	Right-click the setup in the project tree, and clickProperties on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr><tr><td></td><td></td><td></td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditSetup (<SetupName>, <StatisticalParams>)
Python Example	`See EditSetup [Optimization]`

InsertSetup [Statistical]

Inserts a new statistical setup.

UI Access	Right-click **Optimetrics** in the project tree, and then click **Add** > **Statistical** on the shortcut menu.

Parameters

Name	Type	Description
<*StatisticalParams*>	Array	`["NAME:<SetupName>", "SaveFields:=", <SaveField>, <StartingPoint>, "MaxIterations:=", <MaxIter>, "PriorPSetup:=", <PriorSetup>, "PreSolvePSetup:=", <Preceed>, <StatisticalVars>, ["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>], …, ["NAME:Goal", <OptiGoalSpec>]]]`
<*StatisticalVars*>	Array	Parameter of <*StatisticalParams*>: `["NAME:Variables", "VarName:=", ["i:=", <IncludeVar>, "Dist:=", <DistType>, "Tol:=", <Tolerance>, "StdD:=", <StdD>, "Min:=", <MinCutoff>, "Max:=", <MaxCutoff>, ...] "VarName:=", ["i:=", <IncludeVar>, "Dist:=",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<DistType>, "Tol:=", <Tolerance>, "StdD:=", <StdD>, "Min:=", <MinCutoff>, "Max:=", <MaxCutoff>]]
	<DistType>	String	Parameter of <*StatisticalVars*>; distribution can be "Gaussian" or "Uniform".
	<Tolerance>	VarValue	Parameter of <*StatisticalVars*>; the tolerance for the variable when distribution is Uniform
	<StdD>	VarValue	Parameter of <*StatisticalVars*>; the standard deviation for the variable when distribution is Gaussian.
	<MinCutoff>	Double	Parameter of <*StatisticalVars*>; the minimum cut-off for the variable when distribution is Gaussian.
	<MaxCutoff>	Double	Parameter of <*StatisticalVars*>; the maximum cut-off for the variable when distribution is Gaussian.
Return Value	None		

Python Syntax	InsertSetup ("OptiStatistical", <*StatisticalParams*>)
Python Example	oModule.InsertSetup("OptiStatistical", ["NAME:StatisticalSetup1", "SaveFields:=", true, ["NAME:StartingPoint"], "MaxIterations:=", 50, "PriorPSetup:=", "", ["NAME:Variables"], ["NAME:Goals", ["NAME:Goal", "Solution:=", "Setup1 : LastAdaptive",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Calculation:=", "returnloss",
"Context:=", "",
["NAME:Ranges",
"Range:=", ["Var:=", "Freq",
"Type:=", "s",
"Start:=", "8GHz", "Stop:=", "8GHz"]]],
["NAME:Goal",
"Solution:=", "Setup1 : LastAdaptive",
"Calculation:=", "reflect",
"Context:=", "",
["NAME:Ranges",
"Range:=", ["Var:=", "Freq", "Type:=",
"s", "Start:=", "8GHz", "Stop:=", "8GHz"]]]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

12 - Solutions Module Script Commands

Solutions commands should be executed by the "Solutions" module.

```
oModule = oDesign.GetModule("Solutions")
```

```
oModule.CommandName <args>
```

DeleteSolutionVariation

EditSources (Layout Editor)

ExportEigenmodes

ExportNetworkData

FFTOnReport

GetAvailableVariations

SetQ3DSATNetChanges (Layout Editor)

DeleteImportData

Deletes imported solution or table data in HFSS. Not in HFSS-IE

UI Access	Right-click on **Results** > **Import Solutions**, click **Delete Selections** button.		
Parameters	Name	Type	Description
	<ImportSpecArray>	Array	Array of import solutions. `[<ImportName:SolnName OR TableName>, ...]`
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteImportData(<*ImportSpecArray*>)

Python Example	`oModule.DeleteImportData([`
	`  "Import1:Adaptive_1", "Import2:DataTable"`
	`  ])`

DeleteSolutionVariation

Deletes all solution data for specific solutions and design variations. This is obsolete and is supported only for backward compatibility. You should use DeleteFullVariation.

UI Access	Right-click on **Results**, select **Browse Solutions...**, click **Delete** button in the dialog.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SoluParams*></td><td>Array</td><td>Structured array. `[<DataSpecifierArray>, ...]`</td></tr><tr><td><*DataSpecifierArray*></td><td>Array</td><td>Structured array. `[<DesignVariationKey>, <SetupName>, <SolnName>]`</td></tr></table>
Return Value	None.

Python Syntax	DeleteSolutionVariation(<*SoluParams*>)
Python Example	`oModule.DeleteSolutionVariation([` `  ["width='2in'", "Setup1", "Adaptive_1"],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	``` ["width='2in'", "Setup1", "Sweep1"]]) ```

EditQ3DSources (Layout Editor)

Edits the parameters of the Q3D simulation's source excitations.

UI Access	From the **Project Manager** window, right-click a Q3D setup and select **Q3D Edit Sources**.
Parameters	

Name	Type	Description
<Contents>	Array	Structured array containing these parameters in order: • <NAME:EditSourcesCmd> • <Q3D Setup Name> • <N Sources Value Type> • <Sources>
<NAME:EditSourcesCmd>	String	Identifies the contents of the array (i.e., "EditSourcesCmd").
<Q3D Setup Name>	String	Name of the new solver setup (e.g., Q3D1).
<N *Sources Value Type*>	String	Type of the source value, where *N* is any of the following: **A** — current. **V** — voltage. **N** — not defined.
<DC Loss Type>	String	Identifies the DC volume loss calculation assumption. String "True DC Loss" or "DC Loss Low Frequency Sinusoidals"
<Sources>	Array	Structured array containing these parameters in order: • <NAME:Sources> • <IndividualSource>
<NAME:Sources>	String	Identifies the contents of the following arrays (i.e., "Sources").

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<IndividualSource>	Array	Structured array containing these parameters in order: • <NAME:*N* Sources> • <Entry>
	<NAME:N Sources>	String	Identifies the contents of the array (e.g., "AC Sources"), where *N* is **AC**, **DC**, or **CG**.
	<Entry>	Array	Structured array containing some of these parameters in order: • <ID> • <Name> • <Value> • <Magnitude> • <Phase>
	<ID>	Integer	Assigned ID number of the excitation (e.g., 372).)
	<Name>	String	Name of the excitation (e.g., Port1).
	<Value>	String	Value of the excitation (e.g., 0A).
	<Magnitude>	String	Magnitude value of the excitation (e.g., 0V).
	<Phase>	String	Phase value of the excitation (e.g., 0deg).
Return Value	None.		

Python Syntax	EditQ3DSources([<*Contents*>])

Python Example	`oModule.EditQ3DSources(`
	`[`
	`  [`
	`  "NAME:EditSourcesCmd",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Q3D Setup Name:=" , "Q3D1",

"AC Sources Value Type:=", "A",

"CG Sources Value Type:=", "N",

"DC Sources Value Type:=", "A",

"DC Loss Type:=", "DC Loss Low Frequency Sinusoidals",

  [

  "NAME:Sources",

    [

    "NAME:AC Sources",

    "Entry:=", ["ID:=", 372, "Name:=", "Port1", "Magnitude:=",
    "0A","Phase:=", "0deg"],

    "Entry:=", ["ID:=", 366, "Name:=", "Port2", "Magnitude:=",
    "1A","Phase:=", "30deg"],

    "Entry:=", ["ID:=", 370, "Name:=", "Port2#ref", "Value:=",
    "0V","Phase:=", "0deg"],

    "Entry:=", ["ID:=", 374, "Name:=", "Port3#ref", "Value:=",
    "0V","Phase:=", "0deg"]

    ],

    [

    "NAME:CG Sources",

    "Entry:=", ["ID:=", 0, "Name:=", "backplane", "Magnitude:=", "0V",
    "Phase:=", "0deg"],

    "Entry:=", ["ID:=", 2, "Name:=", "RXnet_pin1", "Magnitude:=", "1V",
    "Phase:=", "90deg"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Entry:=", ["ID:=", 1, "Name:=", "RXnet_pin3", "Magnitude:=", "0V",
        "Phase:=", "0deg"]

        ],

        [

        "NAME:DC Sources",

        "Entry:=", ["ID:=", 372, "Name:=", "Port1", "Magnitude:=", "0A",
        "Phase:=", "0deg"],

        "Entry:=", ["ID:=", 366, "Name:=", "Port2", "Magnitude:=", "1A",
        "Phase:=", "0deg"],

        "Entry:=", ["ID:=", 370, "Name:=", "Port2#ref", "Magnitude:=", "0A",
        "Phase:=", "0deg"],

        "Entry:=", ["ID:=", 374, "Name:=", "Port3#ref", "Magnitude:=",
        "0A", "Phase:=", "0deg"]

        ]

     ]

   ]
])
```

EditSources (Layout Editor)

Indicates which source excitations should be used for fields post-processing.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>From the Project Manager window, right-click a Q3D setup and select Q3D Net Settings/Q3D Edit Sources.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><acData></td><td>Array</td><td>Structured array that corresponds to the AC RL tab in the UI. Within this array, each sub-array corresponds to columns in the UI. The Name column is represented by <sourceNames>, and <sourceValues> assigns the Value and Unit to those sources, in the same order.
["NAME:AC",
"Value Type:=" , <string "V" for Voltage or "A" for Current>,
<sourceNames array>, <magnitudeArray>
]</td></tr>
<tr><td><capData></td><td>Array</td><td>Structured array that corresponds to the CG tab in the UI. Within this array, each sub-array corresponds to columns in the UI. The Name column is represented by <sourceNames>. The first <sourceValues> array assigns the Magnitude and its Unit, while the second <sourceValues> array assigns the Phase and its Unit.
["NAME:Cap",
"Value Type:=" , <string "N" for null>,
<sourceNames array>, <magnitudeArray>, <phaseArray>
]</td></tr>
<tr><td><dcData></td><td>Array</td><td>Structured array that corresponds to the DC RL tab in the UI. Within this array, each sub-array corresponds to columns in the UI. The Name column is represented by <sourceNames>, and <sourceValues> assigns the Value and Unit to those sources, in the same order.
["NAME:DC",
"Value Type:=" , <string "V" for Voltage or "A" for Current>,</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<sourceNames array>, <magnitudeArray>]
	<sourceNames>	Array	Structured array. ["NAME:Source Names", <string source name>, <string source name>, ...]
	<magnitudeArray>	Array	Structured array containing magnitude values in the same order as the sources in <sourceNames>. ["NAME:Source Magnitudes", <string containing value and unit>, <string containing value and unit>, ...]
	<phaseArray>	Array	Structured array containing phase values in the same order as the sources in <sourceNames>. ["NAME:Source Phases", <string containing value and unit>, <string containing value and unit>, ...]
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditSources(<*acData*>, <*capData*>, <*dcData*>)
Python Example	oModule.EditSources(["NAME:AC", "Value Type:=" , "A", ["NAME:Source Names", "Port2", "Port2#ref], ["NAME:Source Magnitudes", "0A", "1A"]], ["NAME:Cap", "Value Type:=" , "N", ["NAME:Source Names", "Port1_SINK", "Port1#ref_SINK"], ["NAME:Source Magnitudes", "0V", "1V"], ["NAME:Source Phases", "0deg",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "1deg"
        ]
],
[
"NAME:DC",
"Value Type:=" , "A",
        [
        "NAME:Source Names",
        "Port2",
        "Port2#ref"
        ],
        [
        "NAME:Source Magnitudes",
        "0A",
        "1A"
        ]
])
```

ExportEigenmodes

Exports a tab delimited table of Eigenmodes in HFSS. Not in HFSS-IE.

UI Access	N/A
Parameters	

Name	Type	Description
<*SoluName*>	String	Name of specified solution.
<*DesignVariation*>	String	Design variation key.
<*FileName*>	String	Output file name with path.

Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportEigenmodes(<*SoluName*>, <*DesignVariation*>, <*FileName*>)

Python Example	`oModule.ExportEigenmodes(`
	`"Setup1 : LastAdaptive", "",`
	`"C:\mydir\myeigenmode.eig")`

ExportForHSpice

Exports matrix solution data to a file in a format suitable for HSpice analysis. Available only for Driven Terminal solution types with ports. Output in an appropriate format will be generated for each of the non-empty file names provided.

UI Access: N/A

Parameters

Name	Type	Description
<*DesignVariation*>	String	Desgin variation key.
<*SoluSelections*>	Array	Array of selected solutions.
<*SpiceType*>	Integer	1 - HSpice.
<*BandWidth*>	Integer	0 - Low (narrow) band width.
<*FWSFile*>	String	Full wave spice file name.
<*LumpedElementFile*>	String	Lumped element file name.
<*PoleZeroSpiceFile*>	String	Pole zero spice file name.
<*PoleZeroMatlabFile*>	String	Pole zero Matlab file name.
<*PartialFractionFile*>	String	Partial fraction file name.
<*FittingError*>	Double	The accuracy to use in fitting the pole zero model.
<*MinimumOrder*>	Integer	Minimum number of poles in rational function expansion.
<*MaximumOrder*>	Integer	Maximum number of poles in rational function expansion.
<*UseCommonGround*>	Integer	Optional. 1 to use common ground, 0 otherwise.
<*EnforcePassivity*>	Integer	Optional. 1 to enforce passivity, 0 otherwise.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportForHSpice(<*DesignVariation*>, <*SoluSelections*>, <*SpiceType*>, <*BandWidth*>, <*FWSFile*>, <*LumpedElementFile*>, <*PoleZeroSpiceFile*>, <*PoleZeroMatlabFile*>, <*PartialFractionFile*>, <*FittingError*>, <*MinimumOrder*>, <*MaximumOrder*>, [Optional <*UseCommonGround*>], [Optional <*EnforcePassivity*>])

Python Example	`oModule.ExportForHSpice("width='2in'",`
	`["Setup1:Sweep1"], 1, 0,`
	`"c:\mydir\Sweep1.fws", "", "", "", "",`
	`.005, 20, 200)`

ExportNetworkData

Exports matrix solution data to a file.

UI Access	N/A
Parameters	

Name	Type	Description
<*DesignVariationKey*>	String	Design variation key. Pass empty string for the current nominal variation.
<*SolnSelectionArray*>	Array	Array of selected solutions. `[<SolnSelector>,<SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating sweep.
<*SolnSelector*>	String	Solution setup name and solution name, separated by a colon.
<*FileFormat*>	Integer	File format value.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		2 : Tab delimited spreadsheet format (.tab) 3 : Touchstone (.sNp) 4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<OutFile>	String	Full path to the file to write out.
<FreqsArray>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<DoRenorm>	Boolean	For Touchstone format only. Specifies whether to renormalize the data before export.
<RenormImped>	Double	For Touchstone format only. Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<DataType>	Array	Optional. Type: "S", "Y", or "Z". The matrix to export.
<pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<ComplexFormat>	Integer	Optional. Type: "0", "1", or "2" The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.
<Precision>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
<UseExportFreqs>	Boolean	Specifies whether to use export frequencies.
<IncludeGammaComments>	Boolean	Touchstone only. Specifies whether to include Gamma and Imped-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ance comments.
	<*SupportNonStdExport*>	Boolean	Specifies whether to support non-standard Touchstone extensions for mixed reference impedance.
Return Value	None.		

Python Syntax	ExportNetworkData(<*DesignVariationKey*>, <*SolnSelectionArray*>, <*SolnSelector*>, <*FileFormat*>, <*OutFile*>, <*FreqsArray*>, <*DoRenorm*>, <*RenormImped*>, <*DataType*>, <*pass*>, <*ComplexFormat*>, <*Precision*>, <*UseExportFreqs*>, <*IncludeGammaComments*>, <*SupportNonStdExport*>)
Python Example	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

ExportNMFData

Exports matrix solution data to a file in neutral model format. Available only for Driven solution types with ports. Variables can be held constant by setting their values in the variation field. For example: "length='50mm' width='30mm'". All other independent variables will be treated as NMF parameters.

UI Access	N/A		
Parameters	Name	Type	Description
	<*sourceName*>	Array	Array of selected solutions.
	<*OutFile*>	String	Full path to the file to write out, including the .nmf extenstion
	<*FreqsArray*>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (*e.g.* "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<DesignVariationKey>	String	Design variation.
	<DoRenorm>	Boolean	Specifies whether to renormalize the data before export.
	<RenormImped>	Double	Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
	<Pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
	<ComplexFormat>	Integer	Optional. The format to use for the exported data. Options are 0 = Magnitude/Phase 1= Real/Imaginary. 2= db/Phase.
	<Precision>	Integer	Optional. Number of digits precision. If not specified, the default is 6.
Return Value	None		

Python Syntax	ExportNMFData(<*sourceName*>, <*OutFile*>, <*FreqsArray*>, <*DesignVariationKey*>, <*DoRenorm*>, <*RenormImped*>, <*Pass*>, <*ComplexFormat*>, <*Precision*>)
Python Example	`oModule.ExportNMFData(["Setup1:Sweep1"],` `"c:\mydir\out.nmf", ["all"], "", False, 0)`

FFTOnReport

Performs an FFT on a selected report.

UI Access	Right-click on **Results** in the project tree, select **Perform FFT On Report...**		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*ReportName*>	String	Name of specified report.
	<*WindowName*>	String	Name of window to apply for FFT. Possible values are "Rectangular", "Tri", "Van Hann", "Hamming", "Blackman", "Lanczos", "Weber", "Welch".
	<*Function*>	String	Function to apply on transformation values. Possible values are "none", "ang_deg", "ang_rad", "arg", "cang_deg", "cang_rad", "dB", "dB1 normalize", "dB20normalize", "dBc", "im", "mag", "normalize", "re".
Return Value	None.		

Python Syntax	FFTOnReport (<*ReportName*>, <*WindowName*>, <*Function*>)
Python Example	`oModule.FFTOnReport("XY Plot 1", "Rectangular", "dB")`

GetAdaptiveFreq

Obtains an adaptive frequency for a specified setup.

UI Access	N/A		
Parameters	Name	Type	Description
	<*SetupName*>	String	Name of specified setup.
Return Value	Double frequency value.		

Python Syntax	GetAdaptiveFreq(<*SetupName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.GetAdaptiveFreq("Setup1")`

GetAvailableVariations

Returns the available variation for a solution.

UI Access	N/A
Parameters	Name | Type | Description *<SoluName>* | String | Name of specified solution.
Return Value	Array of strings containing available variations.

Python Syntax	GetAvailableVariations(*<SoluName>*)
Python Example	`oModule.GetAvailableVariations("Setup1 : LastAdaptive")`

GetExcitationScaling

Gets source scaling parameters.

UI Access	N/A
Parameters	Name | Type | Description *<portName>* | String | Port ID. *<Index>* | Integer | Optional. Port/mode/terminal/eigenmode index, if present.
Return Value	Array of strings containing source scaling parameters.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetExcitationScaling(<*portName*>, <*portIndex*>)
Python Example	`oModule.GetExcitationScaling("Port1", 2)`

GetISolutionVersionID

Use: To obtain the solution ID to help track solution validity.

Syntax: GetISolutionVersionID(BSTR fullSolutionName)

Return Value: Returns a solution ID.

Parameters: None

GetSolveRangeInfo

Determines the frequency range of a particular simulation setup. For fast sweeps and interpolating sweeps this command returns the start and stop frequencies. For discrete sweeps, it returns a list of frequencies. For an adaptive solution, it returns the adaptive frequency.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluName*></td><td>String</td><td>Name of specified solution.</td></tr><tr><td><IncludeRangeFieldSaved></td><td>Boolean</td><td>Optional. Specify whether to only include the range with field saved.</td></tr></table>
Return Value	Array of strings containing frequency values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetSolveRangeInfo(<*SoluName*>, [Optional <*IncludeRangeFieldSaved*>])
Python Example	`oModule.GetSolveRangeInfo("Setup1:Sweep1")`

GetSourceContexts

Obtains sources currently enabled as context in the Edit Sources dialog Source Context tab.

UI Access	N/A
Parameters	None.
Return Value	Array of enabled source names

Python Syntax	GetSourceContexts()
Python Example	`oModule.GetSourceContexts()`

GetSourceData (Layout Editor)

Takes a source name as input and returns a VARIANT array containing a data block of the source data.

Command: None

Syntax: GetSourceData

Return Value: Type: Array of i/o block data

Parameters: <SourceName>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

GetValidISolutionList

Gets all available solution names that exist in a design.

UI Access	N/A
Parameters	Name: *<IncludeImportedSolutions>* — Type: Boolean — Description: Optional. Specifies whether to include imported solutions. Default is false.
Return Value	Array of strings containing solution names.

Python Syntax	GetValidISolutionList([Optional *<IncludeImportedSolutions>*])
Python Example	`oModule.GetValidISolutionList()`

HasFields

Determines if fields exist for a particular solution.

UI Access	N/A
Parameters	Name: *<SoluName>* — Type: String — Description: Name of specified solution. Name: *<DesignVar>* — Type: String — Design variation.
Return Value	Integer (1 - True, 0 - False).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	HasFields(<*SoluName*>, <*DesignVar*>)
Python Example	`oModule.HasFields("Setup1:Sweep1", "x_size=2mm")`

HasMatrixData

Determines if matrix data exists for a particular solution.

UI Access	N/A
Parameters	Name | Type | Description <*SoluName*> | String | Name of specified solution. <*DesignVar*> | String | Design variation.
Return Value	Integer (1 - True, 0 - False).

Python Syntax	HasMatrixData(<*SoluName*>, <*DesignVar*>)
Python Example	`oModule.HasMatrixData("Setup1:Sweep1", "")`

HasMesh

Determines if a current mesh exists for a particular simulation setup, not including the initial mesh.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignName*></td><td>String</td><td>Name of specified design.</td></tr><tr><td><*DesignVar*></td><td>String</td><td>Design variation.</td></tr></table>
Return Value	Integer (1 - True, 0 - False).

Python Syntax	HasMesh(<*DesignName*>, <*DesignVar*>)
Python Example	`oModule.HasMesh("Setup1", "")`

ImportSolution

Imports a matrix solution in HFSS, which can then be used in creating reports or in the display of matrix data. The imported solution need not have the same characteristics as the current design. Imported terminal data that meets the required criteria can be used for full-wave Spice export. Not used in HFSS-IE.

UI Access	Right-click on **Results** > **Import Solutions...**, click on **Import Solution...** in the opened dialog window.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Location of the source data. The type of the data file will be determined strictly by its file extension. Supported types are Touchstone (.sNp or .yNp or .zNp or .tou), and Ansoft Designer (.flp).</td></tr><tr><td><*ImportName*></td><td>String</td><td>Identifying name to use for the import, analogous to solution setup name.</td></tr><tr><td><*SoluArray*></td><td>Array</td><td>The names of the solutions selected for import from the file. The only import format supporting multiple solutions in one file is HFSS8.x format.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ImportSolution(<*FileName*>, <*ImportName*>, <*SoluArray*>)
Python Example	`oModule.ImportSolution(` `"c:\mydir\in.s2p", "MeasuredData", ["Sweep1"])`

ImportTable

Imports a data table for use in plotting reports in HFSS. Not used in HFSS-IE. The table can have multiple independent real-valued columns of data, and multiple dependent real- or complex-valued columns of data. The data supported imports are either tab delimited format (.tab) or comma delimited format (.csv). The first row may contain column names. Complex data columns are inferred from the column data format. In tab delimited format, "(double, double)" denotes a complex number. In comma delimited format, "(double, double)" denotes a complex number.

UI Access	Right-click on **Results** > **Import Solutions...**, click on **Import Table...** in the opened dialog window.
Parameters	

Name	Type	Description
<*FileName*>	String	Location of the source data.
<*ImportName*>	String	Identifying name to use for the import, analogous to solution setup name.
<*TableName*>	String	Identifying name for the table, analogous to solution name.
<*IsRealImag*>	Boolean	Whether to use real/imag to interpret data for any complex column. If false, then use mag/phase(degrees).
<*ContainsSoluData*>	Boolean	Whether contains solution data.
<*ColNames*>	Array	Non-empty array used only if intend to override the column names obtained from the table data file, in which case all column names are required.
<*ColIndepFlags*>	Array	Indicates which columns are independent. If this is the empty array, the default is that only the first column is independent. If this is the non- empty array, a flag must be present for every column.

Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ImportTable(<*FileName*>, <*ImportName*>, <*TableName*>, <*IsRealImag*>, <*ContainsSoluData*>, <*ColNames*>, <*ColIndepFlags*>)

Python Example	`oModule.ImportTable("c:\mydir\mytable.tab",`
	`"ImportData", "Measurements", True, True,`
	`[], [True, True, False, False, False])`

IsFieldAvailableAt

Determines if a field solution exists for a particular frequency in a simulation.

UI Access	N/A
Parameters	

Name	Type	Description
<*SoluName*>	String	Name of specified solution.
<*DesignVar*>	String	Design varitaiton.
<*Freq*>	Double	Specified frequency.

Return Value	Integer (1 - True, 0 - False).

Python Syntax	IsFieldAvailableAt(<*SoluName*>, <*DesignVar*>, <*Freq*>)
Python Example	`oModule.IsFieldAvailableAt("Setup1:Sweep1", "", 100.0)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ListMatchingVariations

Gets a list of solved variations that include the specified variable values.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluName*></td><td>String</td><td>Name of specified solutions.</td></tr><tr><td><*MatchVarNames*></td><td>Array</td><td>Array of matching variable names.</td></tr><tr><td><*MatchValStrings*></td><td>Array</td><td>Array of matching variable value strings including units.</td></tr></table>
Return Value	Array of strings corresponding to solved variations. The match variables may be a partial set of design variables and the match values are one per variable in the same order as the variables.

Python Syntax	ListMatchingVariations(<*SoluName*>, <*MatchVarNames*>, <*MatchValStrings*>)

Python Example	`oModule.ListMatchingVariations(`
	`"Setup1:LastAdaptive",`
	`["x_size", "y_size"], ["2mm", "1mm"])`

ListValuesOfVariable

Gets the values of a specified variable corresponding to the solved variations.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluName*></td><td>String</td><td>Name of specified solution.</td></tr><tr><td><VariableName></td><td>String</td><td>Name of specified variable.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Array of double precision values in SI units interpreted as the specified variable corresponding to the solved variations.

Python Syntax	ListValuesOfVariable(<*SoluName*>, <*VariableName*>)
Python Example	`oModule.ListValuesOfVariable("Setup1:LastAdaptive", "x_size")`

SetQ3DSATNetChanges (Layout Editor)

Edits nets in a Q3D setup, including combining and changing net states.

UI Access	From the **Project Manager** window, right-click a Q3D setup and select **Q3D Net Settings**.

Parameters

Name	Type	Description
<SetupName>	String	The name of the setup.
<Q3DSATNetChangesArray>	Array	An array that contains these parameters in order: • <SATFile> • <NetGroupsArray> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<SATFile>	String	Directory path to the SAT file.
<NetGroupsArray>	Array	An array that contains these parameters in order: • <NAME:NetGroups> • <NAMEArray> The parameter <NAMEArray> may repeat, as necessary, for each user-selected change (e.g., changing net states, combining nets, et

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			cetera). This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<NAME:NetGroups>	String	Identifies the contents of the array (i.e., NetGroups).
	<NAMEArray>	Array	An array that contains these parameters in order: • <NAME> • <Name> • <NetArray> • <NetType> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<NAME>	String	Identifies the net or net group being created or modified (e.g., NAME:Nets3).
	<Name>	String	Identifies the net or net group being created or modified (e.g., Nets3).
	<NetsArray>	Array	An array that contains the parameter <NAME:Nets> and then either a list of nets to combine, or a single restatement of the net name.
	<NAME:Nets>	String	Identifies the contents of the array (i.e., Nets).
	<Nets>	String	A list of nets to modify combine, or the name of the net or net group being modified.
	<NetType>	String	The state of the net being created or modified (i.e., REGULAR, FLOATING, or GROUND).
Return Value	None		

Python Syntax	oProject.SetQ3DSATNetChanges(<SetupName>, [<SATFile>, [<NAME:NetGroups>, [<NAME>, <Name>], [<NetType>]]])
Python	`oModule.SetQ3DSATNetChanges("Q3D1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example

```
[
  "SATFile:=" , "C:/User-
  s/USERNAME/AppData/Local/Temp/ZC38E33379616974771197.aedtresults/Test_Design/-
  model.sat",
  [
  "NAME:NetGroups",
  [
    "NAME:Nets3",
    "Name:=" , "Nets3",
    [
      "NAME:Nets",
      "Nets1",
      "Nets2"
    ],
    "NetType:=" , "SIGNAL"
  ],
  [
    "NAME:Nets3",
    "Name:=" , "Nets3",
    [
      "NAME:Nets",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NETS3"
      ],
      "NetType:=" , "FLOATING"
    ],
    ]
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

13 - Field Overlays Module Script Commands

Field overlay commands should be executed by the Field Overlays module, which is called "FieldsReporter" in scripts.

```
oModule = oDesign.GetModule("FieldsReporter")
```

```
oModule.CommandName <args>
```

AddMarker

AddMarkerToPlot

AddNamedExpression

CalcOp

CalcStack

CalculatorRead

CalculatorWrite

ChangeGeomSettings

ChangeProperty (Marker)

ClcEval

ClcMaterial

ClcMaterialValue

ClearAllMarkers

ClearAllNamedExpr

CopyNamedExprToStack

DeleteFieldPlot

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteMarker

DeleteNamedExpr

DeleteUneditablePlot

DoesNamedExpressionExists

EditSurfaceMeshSummaryData

EnterComplex

EnterComplexVector

EnterCoord

EnterEdge

EnterLine

EnterOutputVar

EnterPoint

EnterQty

EnterScalar

EnterScalarFunc

EnterSurf

EnterVector

EnterVectorFunc

EnterVol

ExportFieldPlot

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOnGrid [Field Overlays]

ExportPlotImageToFile

ExportPlotImageWithViewToFile

ExportSurfaceMeshSummary

ExportToFile

GetFieldFolderNames

GetFieldPlotNames

GetFieldPlotQuantityName

GetMeshPlotNames

GetTopEntryValue

HideAntennaParametersOverlay

HideRadiatedPlotOverlay

HidePolarPlot

LoadNamedExpressions

ModifyInceptionParameters

ReassignFieldPlot

RenamePlotFolder

SaveFieldsPlots

SaveNamedExpressions

SetFieldPlotSettings

SetPlotsViewSolutionContext

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SaveFieldsPlots

UpdateAllFieldsPlots

UpdateQuantityFieldsPlots

AddMarkerToPlot

Adds a marker to a trace on a named field plot.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Location*></td><td>Array</td><td>Array of strings containing X.Y, and Z coordinates for the marker.</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of the field plot.</td></tr></table>
Return Value	None

Python Syntax	AddMarkerToPlot(<*Location*>, <*PlotName*>)

Python Example	`oModule.AddMarkerToPlot(`
	`  ["0.290455877780914in", "-0.616900205612183in",`
	`  "1.77635683940025e-015in"],`
	`  "Mag_H1")`
	
	`oModule.AddMarkerToPlot(`
	`  ["-0.317279517650604in",`
	`  "1.22481322288513in", "0in"],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Mag_H1")

AddNamedExpression

Creates a named expression using the expression at the top of the stack.

UI Access	Click **Add** in the **Fields Calculator** window.
Parameters	Name / Type / Description <*ExprName*> / String / Name for the new named expression. <*FieldType*> / String / Type of field. <*Expression*> / Array / Defines an expression to be solved in the calculator.
Return Value	None

Python Syntax	AddNamedExpression(<*ExprName*>, <*FieldType*>)
Python Example	`oModule.AddNamedExpression("Mag_JxE", "Fields")`

```
oModule.AddNamedExpression("new2", "Fields",
```

CalcOp

Performs a calculator operation.

UI Access	Operation commands like **Mag**, **+**, etc.
Parameters	Name / Type / Description <*OpString*> / String / The text on the corresponding calculator button.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	CalcOp(<*OpString*>)
Python Example	`oModule.CalcOp("+")`

CalcStack

Performs an operation on the stack.

UI Access	Stack operation buttons such as **Push** and **Pop**.
Parameters	Name | Type | Description <*OpString*> | String | The text on the corresponding calculator button.
Return Value	None.

Python Syntax	CalcStack(<*OpString*>)
Python Example	`oModule.CalcStack("push")`

CalculatorRead

Gets a register file and applies it to the calculator stack.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Read...** in the **Fields Calculator** dialog.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Path to and including name of input register file.</td></tr><tr><td><*SoluName*></td><td>String</td><td>Specified solution to read in.</td></tr><tr><td><*FieldType*></td><td>String</td><td>Type of specified field.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable names, value pairs.</td></tr></table>
Return Value	None.

Python Syntax	CalculatorRead(<*FileName*>, <*SoluName*>, <*FieldType*>, <*VarArray*>)

Python Example	`oModule.CalculatorRead(`
	`  "c:\example.reg",`
	`  "Setup1: LastAdaptive", "Fields",`
	`  ["Freq:=", "10GHz", "Phase:=", "0deg"])`

CalculatorWrite

Writes contents of top register to file.

UI Access	Click **Write...** in the **Fields Calculator** window.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OutputFilePath*></td><td>String</td><td>Path to and including name of output register file.</td></tr><tr><td><*SoluNameArray*></td><td>Array</td><td>Array of strings containing solution names.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable names, value pairs.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CalculatorWrite(<*OutputFilePath*>, <*SoluNameArray*>, <*VarArray*>)

Python Example	`oModule.CalculatorWrite("C:\test.reg",`
	`  ["Solution:=", "Setup1 : LastAdaptive"],`
	`  ["Freq:=", "1GHz", "Phase:=", "0deg"]`
	`)`

ChangeProperty

Changes the properties of a field plot marker.

UI Access	Select a field plot marker in the marker table.		
Parameters	Name	Type	Description
	<*propertyArgs*>	Array	Structured array.
Return Value	None		

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	`oDesign.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:FieldPlotMarkerDataTab",`
	`  ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:m1"],`
	`  ["NAME:ChangedProps",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          [ "NAME:Name", "Value:=" , "Temp" ]
        ]
     ]
  ])

oDesign.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:FieldPlotMarkerDataTab",
      ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp" ],
      ["NAME:ChangedProps",
         ["NAME:Display Value", "Value:=" , True]
       ]
    ]
  ])

oDesign.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:FieldPlotMarkerDataTab",
      ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp" ],
      ["NAME:ChangedProps",
        ["NAME:Display Name", "Value:=" , False ]
      ]
    ]
  ])

oDesign.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:FieldPlotMarkerDataTab",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
       ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp"],
       ["NAME:ChangedProps",
         ["NAME:Display Name", "Value:=" , True ]
        ]
      ]
  ])


oDesign.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:FieldPlotMarkerDataTab",
       ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp" ],
       ["NAME:ChangedProps",
         ["NAME:Display Units", "Value:=" , True ]
       ]
     ]
  ])


oDesign.ChangeProperty(
   ["NAME:AllTabs",
    ["NAME:FieldPlotMarkerDataTab",
      ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp"],
       ["NAME:ChangedProps",
         ["NAME:Color", "R:=" , 255, "G:=" , 255, "B:=" , 255]
       ]
    ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   ])


oDesign.ChangeProperty(
    ["NAME:AllTabs",
     ["NAME:FieldPlotMarkerDataTab",
       ["NAME:PropServers", "FieldsReporter:FieldsPlotMarker:Temp" ],
       ["NAME:ChangedProps",
         ["NAME:Symbol Size Scale", "Value:=" , "1" ]
         ]
      ]
   ])
```

ChangeGeomSettings

Changes the line discretization setting.

UI Access	In the **Fields Calculator** window, click on **Geom Settings...**
Parameters	Name | Type | Description *<LineDiscr>* | Integer | Line discretization value.
Return Value	None.

Python Syntax	ChangeGeomSettings(*<LineDiscr>*)
Python Example	`oModule.ChangeGeomSettings(500)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClcEval

Evaluates the expression at the top of the stack using the provided solution name and variable values.

UI Access	Click **Eval** in the **Fields Calculator** window.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SoluName*></td><td>String</td><td>Name of specified solution.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable name, value pairs.</td></tr><tr><td><*FieldType*></td><td>String</td><td>Optional. Type of specified field.</td></tr></table>
Return Value	None

Python Syntax	ClcEval(<*SoluName*>, <*VarArray*>, [Optional <*FieldType*>])

Python Example	`oModule.ClcEval("Setup1: LastAdaptive",`
	`  ["Freq:=", "10GHz",`
	`  "Phase:=", "0deg"])`

ClcMaterial

Performs a material operation on the top stack element.

UI Access	Click **Matl...** in the **Fields Calculator** window.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*MatString*></td><td>String</td><td>The material property to apply.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*OpString*>	String	Name of operation. Possible values are "mult", or "div".
Return Value	None.		

Python Syntax	ClcMaterial(<*MatString*>, <*OpString*>)
Python Example	`oModule.ClcMaterial("Permeability (mu)" "mult")`

ClearAllNamedExpr

Clears all user-defined named expressions from the list.

UI Access	Click **ClearAll** in the **Fields Calculator** window.
Parameters	None.
Return Value	None.

Python Syntax	ClearAllNamedExpr()
Python Example	`oModule.ClearAllNamedExpr()`

CopyNamedExprToStack

Copies the named expression selected to the calculator stack.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Select a named expression and then click **Copy to stack**.
Parameters	Name | Type | Description <*ExprName*> | String | Name of the expression to be copied to the top of the stack.
Return Value	None.

Python Syntax	CopyNamedExprToStack(<*ExprName*>)
Python Example	`oModule.CopyNamedExprToStack("Mag_JxE")`

CreateFieldPlot

> **Note:**
>
> Use in conjunction with GetGeometryIdsForNetLayerCombinations and GetGeometryIdsForAllNetLayerCombinations.
>
> - GetGeometryIdsForNetLayerCombinations returns ID numbers of all faces and edges of the active design related to the current target combination of nets and layers.
> - GetGeometryIdsForAllNetLayerCombinations returns ID numbers for all possible combinations of nets and layers it is possible to choose.

Use: Creates a field/mesh plot "Field" or a visual ray trace ("VRT") plot.

Syntax: `CreateFieldPlot <PlotParameterArray> ["Field" | "VRT"]`

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <PlotParameterArray> "Field"

```
Array("NAME:<PlotName>",
   "SolutionName:=", <string>,
   "QuantityName:=", <string>,
   "PlotFolder:=", <string>,
   "UserSpecifyName:=", <int>,
   "UserSpecifyFolder:=", <int>,
   "IntrinsicVar:=", <string>,
   "PlotGeomInfo:=", <PlotGeomArray>,
   "FilterBoxes:=", <FilterBoxArray>,
   <PlotOnPointsSettings>,
   <PlotOnLineSettings>,
   <PlotOnSurfaceSettings>,
   <PlotOnVolumeSettings>)
```

SolutionName:

Name of the solution setup and solution formatted as

"<SolveSetupName>: <WhichSolution>",

where <WhichSolution> can be "Adaptive_<n>", "LastAdaptive", or "PortOnly".

For example: "Setup1 : Adaptive_2"

A space is required on either side of the colon (**:**) character. If omitted, the plot will not be created.

QuantityName:

Type of plot to create. Possible values vary according to the solver.

Mesh plots: "Mesh"

PlotFolder:

Name of the folder to which the plot should be added. (Values vary with the design type.) Some possible values include

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"E Field", "H Field", "Jvol", "Jsurf", "SAR Field",
"Jc","Surface-Loss", "QSurf", "Temperature", "Energy",
"Average-Surface-Loss-Density, "Dielectric_Loss",
"MeshPlots", "Heat Flux", and "Displacement"
```

UserSpecifyName

0 if default name for plot is used, 1 otherwise.

Not needed. <PlotName> will be respected regardless of whether thisflag is set.

UserSpecifyFolder

0 if default folder for plot is used, 1 otherwise.

Not needed. The specified PlotFolder will be respected regardless of whether this flag is set.

IntrinsicVar:

Formatted string that specifies the frequency and phase at which to make the plot.

For example: `"Freq='1GHz' Phase='30deg'"`

PlotGeomInfo:

Creating field plot on selected layer-net pairs:

For example, Maxwell: `"PlotGeomInfo:=", [1,"Volume","ObjList", 2,"LC1_1:Top","LC1_1:Top#L1"]`

HFSS example with "Plot on surface" option is not checked:

```
"PlotGeomInfo:=", [1,"Volume","LayerNets",2, "Top",2,
"net1","net2","Ground",1,"GND"]
```

The command creates field plot on two layer-nets combinations. It starts with type "Volume", and subtype "LayerNets", followed by the number of layer-nets combinations which is 2 in this example.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The two layer-nets combinations are (1) "Top" layer, two nets, net names are "net1" and "net2", *i.e.*, [Top, net1] pair, and [Top, net2] pair. (2) "Ground" layer and one net, net name is "GND", *i.e.*, [Ground, GND] pair.

HFSS example with "Plot on surface" option checked:

```
"PlotGeomInfo:=", [1,"Surface","LayerNetsExtFace",2,
"Top",2,"net1","net2",_"Ground",1,"GND"]
```

The command creates field plot on two layer-nets combinations. It starts with type "Surface", and subtype "LayerNetsExtFace", followed by the number of layer-nets combinations which is 2 in this example.

The two layer-nets combinations are (1) "Top" layer, two nets, net names are "net1" and "net2", i.e., [Top, net1] pair, and [Top, net2] pair. (2) "Ground" layer and one net, net name is "GND", i.e., [Ground, GND] pair.

Limitations:

- No Layer/Nets name is supported for field plot command recording should be able to add it based on this change.
- No Layer/Nets name support for pure Layout design field plot command.

`<PlotGeomArray>`

```
Array(<NumGeomTypes>, <GeomTypeData>, <GeomTypeData>, ...)
```

For example: `Array(4, "Volume", "ObjList", 1, "Box1",`
`"Surface", "FacesList", 1, "12", "Line", 1,`
`"Polyline1", "Point", 2, "Point1", "Point2"`

`<NumGeomTypes>`

Type: <int>

Number of different geometry types (volume, surface, line, point) plotted on at the same time.

`<GeomTypeData>`

```
<GeomType>, <ListType>, <NumIDs>, <ID>, <ID>, ...)
```

`<GeomType>`

Type: <string>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Possible values are "Volume", "Surface", "Line", "Point".

<ListType>

Type: <string>

Possible values are "ObjList", or "FacesList".

These are used for the GeomType of "Line" or "Point".

<NumIDs>

Type: <int>

Number of IDs or object names that will follow.

<ID>

Type: <int> or <string>

ID of a face or name of an object, line, or point on which to plot.

<FilterBoxArray>

Array of names of objects to use to restrict the plot range.

```
Array(<NumFilters>, <ObjName>, <ObjName>, ...)
```

```
Example:Array(1, "Box1")
Example: Array(0) no filtering
```

<PlotOnPointSettings>

```
Array("NAME:PlotOnPointSettings",
   "PlotMarker:=", <bool>,
   "PlotArrow:=", <bool>)
```

<PlotOnLineSettings>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  Array("NAME:PlotOnLineSettings",
    Array("NAME:LineSettingsID",
    "Width:=", <int>,
    "Style:=", <string>),
    "IsoValType:=", <string>,
    "ArrowUniform:=", <bool>,
    "NumofArrow:=", <int>)
```

`Style`

Possible values are: `"Cylinder", "Solid", "Dashdash", "Dotdot", "Dotdash"`

`IsoValType`

Possible values are `"Tone", "Fringe", "Gourard"`

`<PlotOnSurfaceSettings>`

```
  Array("NAME:PlotOnSurfaceSettings",
  "Filled:=", <bool>,
  "IsoValType:=", <string>,
  "SmoothShade:=", <bool>,
  "AddGrid:=", <bool>,
  "MapTransparency:=", <bool>,
  "Transparency:=", <doubl.e>,
  "ArrowUniform:=", <bool>
  "ArrowSpacing:=", <double>
  "GridColor:=", Array(<int>, <int>, <int>)
```

`IsoValType`

Possible values are `"Tone", "Line", "Fringe", "Gourard"`

`GridColor`

Array containing the R, G, B components of the color. Components should be in the range 0 to 255.

`<PlotOnVolumeSettings>`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Array("NAME:PlotOnVolumeSettings",
"PlotIsoSurface:=", <bool>,
"CloudDensity:=", <double>,
"PointSize:=", <int>,
"ArrowUniform:=", <bool>,
"ArrowSpacing:=", <double>)
```

Python Example – E Magnitude Field Plot:

```
oModule.CreateFieldPlot(
[
   "NAME:Mag_E1",
   "SolutionName:="      , "Setup1 : LastAdaptive",
   "QuantityName:="      , "Mag_E",
   "PlotFolder:="        , "E Field1",
   "UserSpecifyName:="   , 0,
   "UserSpecifyFolder:=" , 0,
   "IntrinsicVar:="      , "Freq='1GHz' Phase='0deg'",
   "PlotGeomInfo:=", [1  , "Surface", "FacesList", 1, "7"],
   "FilterBoxes:=", [0],
   [
      "NAME:PlotOnSurfaceSettings",
      "Filled:="         , False,
      "IsoValType:="     , "Fringe",
      "SmoothShade:="    , True,
      "AddGrid:="        , False,
      "MapTransparency:=", True,
      "Transparency:="   , 0,
      "ArrowUniform:="   , True,
      "ArrowSpacing:="   , 0.100000001490116,
      "GridColor:="      , [255, 255, 255]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ]
], "Field")
```

Electron Density Field Plot:

```
oModule = oDesign.GetModule("FieldsReporter")
oModule.CreateFieldPlot(
   [
      "NAME:Electron_Density3",
      "SolutionName:="         , "AIR : RFDischarge",
      "UserSpecifyName:="      , 0,
      "UserSpecifyFolder:="    , 0,
      "QuantityName:="         , "Electron_Density",
      "PlotFolder:="           , "RF Discharge Fields",
      "StreamlinePlot:="       , False,
      "AdjacentSidePlot:="     , False,
      "FullModelPlot:="        , False,
      "IntrinsicVar:="         , "Freq=\'0.20000000000000001GHz\'
                                 GasPressure=\'0.02kPascal\'",
      "PlotGeomInfo:="         , [1,"Surface","FacesList",1,"48"],
      "FilterBoxes:="          , [0],
      [
         "NAME:PlotOnSurfaceSettings",
         "Filled:="            , False,
         "IsoValType:="        , "Tone",
         "AddGrid:="           , False,
         "MapTransparency:="   , True,
         "Refinement:="        , 0,
         "Transparency:="      , 0,
         "SmoothingLevel:="    , 0,
         "ShadingType:="       , 0,
         [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "NAME:Arrow3DSpacingSettings",
            "ArrowUniform:="   , True,
            "ArrowSpacing:="   , 0,
            "MinArrowSpacing:=", 0,
            "MaxArrowSpacing:=", 0
            "GridColor:="      , [255,255,255]
         ],
      ],
   "EnableGaussianSmoothing:=" , False,
   "SurfaceOnly:="             , False
], "Field")
```

HFSS Multipaction SEY Plot

```
# ----------------------------------------

# Script Recorded by Ansys Electronics Desktop Version 2026.1.0

# 15:36:54 Jan 15, 2025

# ----------------------------------------

import ScriptEnv

ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")

oDesktop.RestoreWindow()

oProject = oDesktop.SetActiveProject("JPL_coax_r251")

oDesign = oProject.SetActiveDesign("Coaxial_7-8")

oModule = oDesign.GetModule("FieldsReporter")

oModule.CreateFieldPlot(
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:SEY_Plot1",
  "SolutionName:=" , "300MHz_low : Power",
  "UserSpecifyName:=" , 0,
  "UserSpecifyFolder:=" , 0,
  "QuantityName:=" , "Unknown",
  "PlotFolder:=" , "SEY_Plot",
  "StreamlinePlot:=" , False,
  "AdjacentSidePlot:=" , False,
  "FullModelPlot:=" , False,
  "IntrinsicVar:=" , "PowerMultiplier=\'250\' Time=\'0s\'",
  "PlotGeomInfo:=" , [1,"Surface","FacesList",4,"16","17","18","24"],
  "FilterBoxes:=" , [0],
  [
    "NAME:PlotOnSurfaceSettings",
    "Filled:=" , False,
    "IsoValType:=" , "Tone",
    "AddGrid:=" , False,
    "MapTransparency:=" , True,
    "Refinement:=" , 0,
    "Transparency:=" , 0,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "SmoothingLevel:=" , 0,
      "ShadingType:=" , 0,
      [
        "NAME:Arrow3DSpacingSettings",
        "ArrowUniform:=" , True,
        "ArrowSpacing:=" , 0,
        "MinArrowSpacing:=" , 0,
        "MaxArrowSpacing:=" , 0
      ],
    "GridColor:=" , [255,255,255]
  ],
  "EnableGaussianSmoothing:=", False,
  "SurfaceOnly:=" , False,
  "QuantityName:=" , "QuantityName_SecondaryEmission",
  "PlotFolder:=" , "SEY_Plot",
  "IntrinsicVar:=" , "PowerMultiplier=\'250\' Time=\'0s\'"
  ], "SecondaryEmission")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteFieldPlot

Deletes one or more field plots.

UI Access	Right-click on one filed plot, select **Delete**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NameArray*></td><td>Array</td><td>Array of strings containing the names of the plots to delete.</td></tr></table>
Return Value	None.

Python Syntax	DeleteFieldPlot(<*NameArray*>)
Python Example	`oModule.DeleteFieldPlot(["Mag_E1", "Vector_E1"])`

DeleteMarker[Fields Reporter]

Deletes one or more marker in the current field plot.

UI Access	Right-click **Field Overlays** > **Plot Fields** > **Marker** > **Delete Marker**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*MarkerNames*></td><td>Array</td><td>Array of strings containing marker names.</td></tr></table>
Return Value	None.

Python Syntax	DeleteMarker(<*MarkerNames*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.DeleteMarker(["m1", "m2"])`

DeleteNamedExpr

Deletes the selected named expression from the list.

UI Access	Select a named expression and then click **Delete**.
Parameters	Name / Type / Description *<ExprName>* / String / Name of specified named expression.
Return Value	None.

Python Syntax	DeleteNamedExpr(*<ExprName>*)
Python Example	`oModule.DeleteNamedExpr("Mag_JxE")`

DeleteUneditablePlot

Deletes one or more uneditable plots.

UI Access	N/A
Parameters	Name / Type / Description *<NameArray>* / Array / Array of the plot names to delete.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteUneditablePlot(<*NameArray*>)
Python Example	`oModule.DeleteUneditablePlot(["Mag_E1", "Vector_E1"])`

DoesNamedExpressionExists

Determines whether specified named expression exists.

UI Access	N/A
Parameters	Name | Type | Description <*ExpName*> | String | Name of the specified named expression.
Return Value	Boolean: • **1** - named expression exists. • **0** - named expression does not exist.

Python Syntax	DoesNamedExpressionExists(<*ExpName*>)
Python Example	`oModule.DoesNamedExpressionExists("Mag_JxE")`

EditSurfaceMeshSummaryData

Defines or edits Surface Mesh Summary Data.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>HFSS > Fields > Create Surface Mesh Summary, then click the Setup button</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Name></td><td>String</td><td>"NAME:SurfaceMeshSummary"</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Solution name that has the mesh to be evaluated</td></tr>
<tr><td><Variation Name></td><td>String</td><td>Name of design variation</td></tr>
<tr><td><MeshRowItems></td><td>Array</td><td>Row data that includes <ItemPerRow>, <EntityPerRow>, and <MeshDataPer Item></td></tr>
<tr><td><ItemPerRow></td><td>String</td><td>"NAME:ItemPerRow"</td></tr>
<tr><td><EntityPerRow></td><td>String</td><td>Entity ID</td></tr>
<tr><td><MeshDataPer Item></td><td>Array</td><td>Mesh data category and statistics</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>EditSurfaceMeshSummaryData ([[<Name>,<SolutionName>,<Variation Name>,<MeshRowItems>,[<ItemPer-Row>,<EntityPerRow>,<MeshDataPer Item>]...]])</td></tr>
<tr><td>Python Example</td><td>

```
oModule = oDesign.GetModule("FieldsReporter")
oModule.EditSurfaceMeshSummaryData(
  [
  [
    "NAME:SurfaceMeshSummary",
    "SolutionName:="        , "Setup1 : LastAdaptive",
    "Variation:="           , "Nominal",
    [
    "NAME:MeshRowItems",
    [
      "NAME:ItemPerRow",
        "EntityPerRow:="        , "annular_rng",
    [
```

</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:MeshDataPerItem",
        "Min Edge Len:="        , 0.000166961133900917,
        "Max Edge Len:="        , 0.00145730991671888,
        "Mean Edge Len:="       , 0.000524160923260581,
        "Std Devn Edge Len:="   , 0.000217758706109324,
        "Min Tri Area:="        , 3.4982570283924E-09,
        "Max Tri Area:="        , 3.73116346661249E-07,
        "Mean Tri Area:="       , 8.91526236529066E-08,
        "Std Devn Tri Area:="   , 6.89482807990471E-08,
        "DataState:="           , 2
      ]
    ],
    [
      "NAME:ItemPerRow",
        "EntityPerRow:="        , "gap",
      [
        "NAME:MeshDataPerItem",
        "Min Edge Len:="        , 0.000105681286042456,
        "Max Edge Len:="        , 0.000362084671538928,
        "Mean Edge Len:="       , 0.000233090954707978,
        "Std Devn Edge Len:="   , 7.55387369793154E-05,
        "Min Tri Area:="        , 1.74640378345369E-09,
        "Max Tri Area:="        , 3.78166789228631E-08,
        "Mean Tri Area:="       , 1.56497981245286E-08,
        "Std Devn Tri Area:="   , 1.04691637802914E-08,
        "DataState:="           , 2
      ]
    ]
    ]
    ]
])
```

EnterComplex

Enters a complex number onto the stack.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Number**, and then click **Scalar**. **Complex** option is selected.
Parameters	Name | Type | Description <*ComplexNum*> | String | String of complex value. Ex. "1 + 2j".
Return Value	None.

Python Syntax	EnterComplex(<*ComplexNum*>)
Python Example	`oModule.EnterComplex("1 + 2 j")`

EnterComplexVector

Enters a complex vector onto the stack.

UI Access	Click **Number**, and then click **Vector**. **Complex** option is selected.
Parameters	Name | Type | Description <*ComplexVector*> | Array | Array of strings containing X, Y and Z complex values.
Return Value	None.

Python Syntax	EnterComplexVector(<*ComplexVector*>)
Python Example	`oModule.EnterComplexVector` `([`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"1 + 2 j", "1 + 2 j", "1 + 2 j"])

EnterCoord

Enters a coordinate system defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Coord**.
Parameters	Name / Type / Description <*CoordName*> / String / Name of a coordinate system defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterCoord(<*CoordName*>)
Python Example	`oModule.EnterCoord("Global")`

EnterEdge

Enters an edge defined in the 3D Modeler editor.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description <*EdgeName*> | String | Name of an edge defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterEdge(<*EdgeName*>)
Python Example	`oModule.EnterEdge("Edge_1")`

EnterLine

Enters a line defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Line**
Parameters	Name | Type | Description <*LineName*> | String | Name of a line defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterLine(<*LineName*>)
Python Example	`oModule.EnterLine("Line1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnterOutputVar

Enters Output Vars, only valid for Eigenmode problems.

UI Access	Click **Geometry** and then select **Output Vars**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*VarName*></td><td>String</td><td>Name of the output vars. Only 'freq' is supported.</td></tr><tr><td><*VarType*></td><td>String</td><td>Type of the output vars. 'Complex' type.</td></tr></table>
Return Value	None.

Python Syntax	EnterOutputVar(<*VarName*>, <*VarType*>)
Python Example	`oModule.EnterOutputVar("Freq", "Complex")`

EnterPoint

Enters a point defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Point**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PointName*></td><td>String</td><td>Name of a point defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnterPoint(<*PointName*>)
Python Example	`oModule.EnterPoint("Point1")`

EnterQty

Enters a field quantity.

UI Access	Click **Quantity**, and then select from the list.
Parameters	Name | Type | Description <*FieldQty*> | String | The field quantity to be entered onto the stack.
Return Value	None.

Python Syntax	EnterQty(<*FieldQty*>)
Python Example	`oModule.EnterQty("E")`

EnterScalar

Enters a scalar onto the stack.

UI Access	Click **Number** and then click **Scalar**. **Complex** option not selected.
Parameters	Name | Type | Description <*Scalar*> | Double | The real number to enter onto the stack.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	EnterScalar(<*Scalar*>)
Python Example	`oModule.EnterScalar(3.14159265358979)`

EnterScalarFunc

Enters a scalar function.

UI Access	Click **Function** and then select **Scalar**.
Parameters	Name | Type | Description <*VarName*> | String | Name of a variable to enter as a scalar function onto the stack.
Return Value	None.

Python Syntax	EnterScalarFunc(<*VarName*>)
Python Example	`oModule.EnterScalarFunc("Phase")`

EnterSurf

Enters a surface defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Surface**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SurfName*></td><td>String</td><td>Name of a surface defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Python Syntax	EnterSurf(<*SurfName*>)
Python Example	`oModule.EnterSurf("Rectangle1")`

EnterVector

Enters a vector onto the stack.

UI Access	Click **Number**, and then click **Vector**. **Complex** option not selected.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*VectorArray*></td><td>Array</td><td>Array of strings containing X, Y and Z components of the vector.</td></tr></table>
Return Value	None.

Python Syntax	EnterVector(<*VectorArray*>)
Python Example	`oModule.EnterVector([1.0, 1.0, 1.0])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnterVectorFunc

Enters a vector function.

UI Access	Click **Function** and then select **Vector**.
Parameters	Name | Type | Description <*VarNames*> | Array | Array of strings containing names of a variable for the X, Y, and Z coordinates, respectively, to enter as a vector function on the stack.
Return Value	None.

Python Syntax	EnterVectorFunc(<*VarNames*>)
Python Example	`oModuleEnterVectorFunc(["X", "Y", "Z"])`

EnterVol

Enters a volume defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Volume**.
Parameters	Name | Type | Description <*VolumeName*> | String | Name of a volume defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterVol(<*VolumeName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.EnterVol("Box1")`

ExportFieldPlot

Exports a field plot to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PlotName*>	String	Name of the field plot to export.
	<*ShowHeader*>	Boolean	**True** - export field plot header to a file. **False** - export field plot.
	<*FileName*>	String	Name of file to save as, including the file path.
Return Value	None.		

Python Syntax	ExportFieldPlot(<*PlotName*>, <*ShowHeader*>, <*FileName*>)

Python Example	oDefinitionEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "MySymbol")
	`oModule.ExportFieldPlot("Mag_E1",`
	`True, "C:/field_report.dsp")`

ExportMarkerTable

Exports the marker table to a .csv or .tab file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	[product] > **Fields** > **Plot Fields** > **Marker** > **Export Marker Table**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Name of export file include path.</td></tr></table>
Return Value	None.

Python Syntax	ExportMarkerTable(<*FileName*>)
Python Example	`oModule.ExportMarkerTable("C:/work/FieldMarkerTable.csv")`

ExportOnGrid [Field Overlays]

Evaluates the top stack element at a set of points specified by a grid and exports the data to a file.

UI Access	Click **Export**, and then click **On Grid**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OutputFile*></td><td>String</td><td>Name of the output file.</td></tr><tr><td><*Min*></td><td>Array</td><td>Minimum values for the coordinate components of the grid system</td></tr><tr><td><*Max*></td><td>Array</td><td>Maximum values for the coordinate components of the grid system</td></tr><tr><td><*Spacing*></td><td>Array</td><td>Spacing values for the coordinate components of the grid system</td></tr><tr><td><*SolnName*></td><td>String</td><td>Name of the simulation setup</td></tr><tr><td><*SolnParameters*></td><td>Array</td><td>Array of strings containing setup definitions.</td></tr><tr><td><*ExportOption*></td><td>Array</td><td>["NAME:ExportOption" <*IncludePointsInOutput*>: Boolean; specifies whether include points in the output file. <*RefCSName*>: String, specifies the reference coordinate system used by the model</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<*PtInSI*>: Boolean; if True, points are exported in SI. If False, points are exported in model units <*FieldInRefCS*>: Boolean, if True, the points are exported in the reference coordinate system.
	<*CSType*>	String	Optional. Type of coordinate system. "Cartesian" (default) | "Cylindrical" | "Spherical"
	<*Offsets*>	Array	Optional. Origin for the offset coordinate system. For Cartesian, x, y, z, for Cylindrical, R, Phi, Z, for Spherical, Rho, Theta, Phi.
	<*ByCount*>	Boolean	Optional.
Return Value	None.		

Note: Regarding the **ExportOnGrid** legacy script, which only has “IncludePtInOutput” argument, AEDT can still read it and assign other new arguments as default values. Those default values are RefCSName = “global”, PtInSI = “True”, FieldInRefCS = “False”

Python Syntax	ExportOnGrid(<*OutputFile*>, <*Min*>, <*Max*>, <*Spacing*>, <*SolnName*>, <*SolnParameters*>, <*ExportOption*>, "<*CSType*>", <*Offsets*>, <*ByCount*>)

Python Example	`oModule.ExportOnGrid("C:\offset_grid_model_unit_ref.fld",`
	`  ["-1mm", "16mm", "0mm"],`
	`  ["1mm", "18mm", "1mm"],`
	`  ["2mm", "2mm", "1mm"],`
	`  "4500MHz : LastAdaptive",`
	`  ["Freq:=", "4.5GHz","Phase:=" , "0deg"],`
	`  ["NAME:ExportOption",`
	`  "IncludePtInOutput:=" , True,`
	`  "RefCSName:=" , "offset",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"PtInSI:=" , False, "FieldInRefCS:=" , True], "Cartesian", ["0mm", "0mm", "0mm"], False)

ExportPlotImageToFile

Deprecated. Use ExportPlotImageWithViewToFile.

Creates field plot exports of existing field plots from a given view points, and with the model being auto-sized automatically for each view.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Full path plus file name.</td></tr><tr><td><*FolderName*></td><td>String</td><td>Plot folder name.</td></tr><tr><td><*ItemName*></td><td>String</td><td>Name of fields to plot.</td></tr><tr><td><*SetViewTopDownDirectionByRCS*></td><td>String</td><td>Optional. Name of relative coordinate system to use for the field plot.</td></tr></table>
Return Value	None.

Python Syntax	ExportPlotImageToFile(<*FileName*>, <*FolderName*>, <*ItemName*>, <*SetViewTopDownDirectionByRCS*>)
Python Example	oModule.ExportPlotImageToFile("C:\TestEPITF2.jpg", "", "Mag_E2", "RelativeCS1")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportPlotImageWithViewToFile [Reporter]

Exports a field plot image to a specified file.

> **Note:**
>
> This script replaces ExportPlotImageToFile, which is deprecated.

UI Access	**N/A**		
Parameters	Name	Type	Description
	<*FileName*>	String	Full path of file to export.
	<*PlotQuantityName*>	String	Name of quantity to plot.
	<*PlotItemName*>	String	Name of fields to plot.
	<*PixelSizeX*>	Integer	Desired image width, in pixels.
	<*PixelSizeY*>	Integer	Desired image height, in pixels.
	<*ViewOrientation*>	String	*Optional*. Name of orientation to use for plot.
Return Value	Image file is exported to the specified path.		

Python Syntax	ExportPlotImageWithViewToFile(<*FileName*>, <*PlotQuantityName*>, <*PlotItemName*>, <*PixelSizeX*>, <*PixelSizeY*>, <*ViewOrientation*>)
Python Example	`oModule.ExportPlotImageWithViewToFile("E:/MyDir/OptimToutput/magE2.gif", "E Field", "Mag_E2", 1920, 1080, "newot2")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportSurfaceMeshSummary

Use: Exports a Surface Mesh Summary.

Command: **Create Surface Mesh Summary**.

Syntax: ExportSurfaceMeshSummary Array(<SolutionName>, <DesignVariationKey> <ExportFileName>)

Return Value: None

Parameters: `<SolutionName>`

Type: <string>

Original name of the plot.

`<DesignVariationKey>`

Type: <string>

New name of the plot.

Python Syntax	ExportSurfaceMeshSummary (<SolutionName>, <DesignVariationKey>, <ExportFileName>)

Python Example	`oModule.ExportSurfaceMeshSummary(`
	`  [`
	`  "SolutionName:=" , "Setup1 : LastAdaptive",`
	`  "DesignVariationKey:=" , "Nominal",`
	`  "ExportFileName:=" , "D:\\Program Files\\Documents\surfaceMeshSummaryReport.csv"`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportToFile

Note:

The ExportToFile script command has replaced the script command ExportReport. ExportReport remains in order to retain backward compatibility for existing scripts, but it is strongly recommended that you now use ExportToFile.

From a data table or plot, generates text format, comma delimited, tab delimited, or .dat type output files.

<table>
<tr><td>UI Access</td><td>Right-click on report name in the project tree and select Export Data.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><ReportName></td><td>String</td><td>Name of report to be exported.</td></tr>
<tr><td><FileName></td><td>String</td><td>Full path of the exported image file name; with extension of
.txt - Post processor format file
.csv - Comma-delimited data file
.tab - Tab-separated file
.dat - Ansys plot data file</td></tr>
<tr><td><UnitSpec></td><td>String</td><td>For example, “kV, Mhz, yd”</td></tr>
<tr><td><UseTraceNumberFormat></td><td>Boolean</td><td>"True", "False"</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportToFile(<*ReportName*>, <*FileName*>)
Python Example	`oModule.ExportToFile("rETotal", "C:/Users/Documents/rETotal.csv")` `oModule.ExportToFile("S Parameter Table 1", "D:/Users/Documents/cftt.csv", False, " kV, MHz, yd ", True)`

GetFieldFolderNames

Gets the folder names of the field plots.

UI Access	N/A
Parameters	None.
Return Value	Array of folder names.

Python Syntax	GetFieldFolderNames()
Python Example	`oModule.GetFieldFolderNames()`

GetFieldPlotNames

Gets the names of field overlay plots defined in a design.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing field plots names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetFieldPlotNames()
Python Example	`oModule.GetFieldPlotNames()`

GetFieldPlotQuantityName

Gets the quantity name of a specified field plot.

UI Access	N/A
Parameters	Name | Type | Description <*PlotName*> | String | Name of specified plot.
Return Value	String plot quantity name.

Python Syntax	GetFieldPlotQuantityName(<*PlotName*>)
Python Example	`oModule.GetFieldPlotQuantityName("Mag_H1")`

GetMeshPlotNames

Gets the names of mesh plots defined in a design.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None..
Return Value	Array of plot name.

Python Syntax	GetMeshPlotNames
Python Example	`oModule.GetMeshPlotNames()`

GetTopEntryValue

Gets the value of the top entry of the calculator stack.

UI Access	N/A
Parameters	Name | Type | Description <*SolnName*> | String | Name of specified solution. <*VarVals*> | Array | Array of variable name, value pairs.
Return Value	Array of strings containing top entry values.

Python Syntax	GetTopEntryValue(<*SolnName*>, <*VarVals*>)

Python Example	`oModule.GetTopEntryValue(`
	`  "Setup1:LastAdaptive",`
	`  ["Freq:=", "1GHz", "Phase:=", "0deg", "x_size:=", "2mm"]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HideAntennaParameterOverlay

Hides a currently visible antenna parameter overlay.

UI Access	**HFSS** or **HFSS-IE** > **Fields** > **Plot Fields** > **Radiation Field**
Parameters	Name | Type | Description <*Name*> | String | The name of the antenna parameters overlay.
Return Value	None.

Python Syntax	HideAntennaParameterOverlay(<*Name*>)
Python Example	`oModule.HideAntennaParameterOverlay(` `"Antenna Parameter Overlay1")`

HidePolarPlot

Hides a currently visible polar plot.

UI Access	N/A
Parameters	Name | Type | Description <*PlotName*> | String | Name of the polar plot.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	HidePolarPlot(<*PlotName*>)
Python Example	`oModule.HidePolarPlot("rE Plot 1")`

HideRadiatedPlotOverlay

Hides overlay radiation fields plot.

UI Access	N/A
Parameters	Name | Type | Description <*PlotName*> | String | Name of specified plot.
Return Value	None.

Python Syntax	HideRadiatedPlotOverlay(<*PlotName*>)
Python Example	`oModule.HideRadiatedPlotOverlay("Gain Plot 1")`

LoadNamedExpressions

Loads a named expression definition from a saved file.

UI Access	In the Fields Calculator, click **Load From...** in the Library area.
Parameters	Name | Type | Description <*FileName*> | String | Filename and full path to the file to hold the named expression definition.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*FieldType*>	String	For products with just one filed type, it is set to "Fields".
	<*NamedExpr*>	Array	rray of strings containing the names of expression definitions to load from the file.
Return Value	None.		

Python Syntax	LoadNamedExpressions(<*FileName*>, <*FieldType*>, <*NamedExpr*>)

Python Example	`oModule.LoadNamedExpressions(`
	`  "C:\Ansoft\PersonalLib\smth.clc",`
	`  "Fields", ["smoothedtemp"])`

ModifyFieldPlot

Modifies a plot definition.

UI Access	[product] > **Fields** > **Modify Plot**		
Parameters	Name	Type	Description
	<*OriginalName*>	String	Name of the plot to be modified.
	<*PlotParams*>	Array	Array containing modified settings.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ModifyFieldPlot(<*OriginalName*>, <*PlotParams*>)
Python Example	oModule.ModifyFieldPlot("Vector_E1" , ["NAME:Vector_E2", "SolutionName:=", "Setup1 : LastAdaptive", "QuantityName:=", "Vector_E", "PlotFolder:=", "E Field1", "UserSpecifyName:=", 0, "UserSpecifyFolder:=", 0, "IntrinsicVar:=","Freq='1GHz' Phase='30deg'", "PlotGeomInfo:=", [1, "Surface","FacesList", 1, "7"], "FilterBoxes:=", [0], ["NAME:PlotOnSurfaceSettings", "Filled:=", False, "IsoValType:=", "Fringe", "SmoothShade:=", True, "AddGrid:=", False, "MapTransparency:=", True, "Transparency:=", 0, _ "ArrowUniform:=", True, "ArrowSpacing:=", 0.100000001490116, "GridColor:=", [255, 255, 255]]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

ReassignFieldPlot

Reassigns a field plot.

UI Access	Right-click on a field plot > **Reassign**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of specified plot to reassign.</td></tr><tr><td><*PlotParameterArray*></td><td>Array</td><td>See CreateFieldPlot</td></tr></table>
Return Value	None.

Python Syntax	ReassignFieldPlot(<*PlotName*>, <*PlotParameterArray*>)
Python Example	oModule.ReassignFieldPlot "Mag_H1", ["NAME:Mag_H1", "SolutionName:=", "Setup1 : LastAdaptive", "UserSpecifyName:=", 0, "UserSpecifyFolder:=", 0, "QuantityName:=", "Mag_H", "PlotFolder:=", "H Field", _ "StreamlinePlot:=", False "AdjacentSidePlot:=", False, "FullModelPlot:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "IntrinsicVar:=", "Freq=\'20GHz\' Phase=\'0deg\'",
    "PlotGeomInfo:=", [1,"Surface","FacesList",1,"117"
  ],
  "FilterBoxes:=", [0],
  ["NAME:PlotOnSurfaceSettings",
    "Filled:=", False,
    "IsoValType:=", "Fringe",
    "AddGrid:=", False,
    "MapTransparency:=", True,
    "Refinement:=", 0,
    "Transparency:=", 0,
    "SmoothingLevel:=", 0,
    "ShadingType:=", 0,
    ["NAME:Arrow3DSpacingSettings",
      "ArrowUniform:=", True,
      "ArrowSpacing:=", 0,
      "MinArrowSpacing:=", 0,
      "MaxArrowSpacing:=", 0
    ],
    "GridColor:=", [255,255,255]
  ],
  "EnableGaussianSmoothing:=", False
  ])
```

RenameFieldPlot

Renames a plot.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click the plot you want to rename in the project tree, and then click **Rename** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OldName*></td><td>String</td><td>Original name of the plot.</td></tr><tr><td><*NewName*></td><td>String</td><td>New name of the plot.</td></tr></table>
Return Value	None.

Python Syntax	RenameFieldPlot(<*OldName*>, <*NewName*>)
Python Example	`oModule.RenameFieldPlot(` `"Vector_E1", "Vector_E2")`

RenamePlotFolder

Renames a plot folder.

UI Access	Right-click a plot folder in the project tree, and then click **Rename** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OldName*></td><td>String</td><td>Original name of the folder.</td></tr><tr><td><*NewName*></td><td>String</td><td>New name of the folder.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	RenamePlotFolder(<*OldName*>, <*NewName*>)
Python Example	`oModule.RenamePlotFolder(` `"E Field", "Surface Plots")`

SaveFieldsPlots

Saves field plot(s) to a file.

UI Access	**HFSS** > **Fields** > **Save As...**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotNames*></td><td>Array</td><td>Array of plot names.</td></tr><tr><td><*FileName*></td><td>String</td><td>Name of the file to save as, including the file path.</td></tr></table>
Return Value	None.

Python Syntax	SaveFieldsPlots(<*PlotNames*>, <*FileName*>)
Python Example	`oModule.SaveFieldsPlots(["Mag_E1", "Mag_E2"],` `"C:/field_report.dsp")`

SaveNamedExpressions

Saves a named expression definition to a file.

UI Access	In the Fields Calculator, click **Save To...** in the Library area.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Filename and full path to the file to hold the named expression definition.</td></tr><tr><td><NamedExprs></td><td>Array</td><td>Array of strings containing the names of expression definitions to load from the file.</td></tr><tr><td><*Overwrite*></td><td>Boolean</td><td>Specifies whether to overwrite the file.</td></tr></table>
Return Value	None.

Python Syntax	SaveNamedExpressions(<*FileName*>, <*NamedExprs*>, <*Overwrite*>)

Python Example	`oModule.SaveNamedExpressions(`
	`"C:\Ansoft\PersonalLib\smth.clc",`
	`["smoothedtemp"], True)`

SetFieldPlotSettings

Sets plot attributes.

UI Access	[product] > **Fields** > **Modify Plot Attributes**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of the plot to modify.</td></tr><tr><td><*PlotItemAttributes*></td><td>Array</td><td>Structured array. `["NAME:FieldsPlotItemSettings",` `<PlotOnPointsSettings>,`</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<PlotOnLineSettings>, <PlotOnSurfaceSettings>, <PlotOnVolumeSettings>] See description of CreateFieldPlot command for details.
Return Value	None.		

Python Syntax	SetFieldPlotSettings(<*PlotName*> <*PlotItemAttributes*>)
Python Example	oModule.SetFieldPlotSettings("Mag_E2", ["NAME:FieldsPlotItemSettings", ["NAME:PlotOnLineSettings", ["NAME:LineSettingsID", "Width:=", 4, "Style:=", "Cylinder"], "IsoValType:=", "Tone", "ArrowUniform:=", True, "NumofArrow:=", 100], ["NAME:PlotOnSurfaceSettings", "Filled:=", False, "IsoValType:=", "Tone",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "SmoothShade:=", True,
        "AddGrid:=", False,
        "MapTransparency:=", True,
        "Transparency:=", 0,
        "ArrowUniform:=", True,
        "ArrowSpacing:=", 0.100000001490116,
        "GridColor:=", [255, 255, 255]
    ]
])
```

SetPlotFolderSettings

Sets the attributes of all plots in the specified folder.

UI Access	[product] > **Fields** > **Modify Plot Attributes**		
Parameters	Name	Type	Description
	<PlotFolderName>	String	Name of the folder with the attributes to modify.

	<PlotFolderAttributes>	Array	Structured array.
			`["NAME:FieldsPlotSettings",`
			`"Real time mode:=", <bool>,`
			`<ColorMapSettings>,`
			`<Scale3DSettings>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		<Marker3DSettings>, <Arrow3DSettings>]
<ColorMapSettings>	Array	Structured array. ["NAME:ColorMapSettings", "ColorMapType:=", <string Possible values are "Uniform", "Ramp", "Spectrum">, "SpectrumType:=", <string Possible values are "Rainbow", "Temperature", "Magenta", "Gray">, "UniformColor:=", <array containing the R, G, B components of the color. Components should be in the range 0 to 255.>, "RampColor:=", <array containing the R, G, B components of the color. Components should be in the range 0 to 255.>]
<Scale3DSettings>	Array	Structured array. ["NAME:Scale3DSettings", "m_nLevels:=", <int>, "m_autoScale:=", <bool>, "minvalue:=", <double>, "maxvalue:=", <double>, "log:=", <bool>, "IntrinsicMin:=", <double>, "IntrinsicMax:=", <double>]
<Marker3DSettings>	Array	Structured array. ["NAME:Marker3DSettings", "MarkerType:=", <integer>, #9:Sphere

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			#10: Box #11: Tetrahedron #12: Octahedron #default: Sphere "MarkerMapSize:=", <bool>, "MarkerMapColor:=", <bool>, "MarkerSize:=", <double>]
	<Arrow3DSettings>	Array	Structured array. ["NAME:Arrow3DSettings", "ArrowType:=", <integer>, #0: Line #1: Cylinder #2: Umbrella #default: Line "ArrowMapSize:=", <bool>, "ArrowMapColor:=", <bool>, "ShowArrowTail:=", <bool>, "ArrowSize:=", <double>]
Return Value	None.		

Python Syntax	SetPlotFolderSettings(<*PlotFolderName*>, <*PlotFolderAttributes*>)
Python Example	oModule.SetPlotFolderSettings("E Field1", ["NAME:FieldsPlotSettings", "Real time mode:=", True,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:ColorMapSettings",
"ColorMapType:=", "Spectrum",
"SpectrumType:=", "Rainbow",
"UniformColor:=", [127, 255, 255],
"RampColor:=", [255, 127, 127]],
["NAME:Scale3DSettings",
"m_nLevels:=", 27,
"m_autoScale:=", True,
"minvalue:=", 9.34379863739014,
"maxvalue:=", 13683.755859375,
"log:=", False,
"IntrinsicMin:=", 9.34379863739014,
"IntrinsicMax:=", 13683.755859375),
["NAME:Marker3DSettings",
"MarkerType:=", 0,
"MarkerMapSize:=", True,
"MarkerMapColor:=", False,
"MarkerSize:=", 0.25],
["NAME:Arrow3DSettings",
"ArrowType:=", 1,
"ArrowMapSize:=", True,
"ArrowMapColor:=", True,
"ShowArrowTail:=", True,
"ArrowSize:=", 0.25]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ShowAntennaParameterOverlay

Displays an existing antenna parameters overlay as a table on a far field plot.

UI Access	**HFSS** or **HFSS-IE** > **Fields** > **Plot** > **Radiation Fields**
Parameters	Name | Type | Description <*Name*> | String | The name of the antenna parameters overlay.
Return Value	None.

Python Syntax	ShowAntennaParamedterOverlay(<*Name*>)
Python Example	`oModule.ShowAntennaParameterOverlay(` `"Antenna Parameter Overlay1")`

UpdateAllFieldsPlots

Updates the All Fields Plots.

UI Access	N/A
Parameters	None.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	UpdateAllFieldsPlots()
Python Example	`oModule.UpdateAllFieldsPlots()`

UpdateQuantityFieldsPlots

Updates the Quantity Fields Plots.

UI Access	N/A
Parameters	Name | Type | Description *<FolderName>* | String | Folder containing the plotted quantities.
Return Value	None.

Python Syntax	UpdateQuantityFieldsPlots(*<FolderName>*)
Python Example	`oModule.UpdateQuantityFieldsPlots("fldplt")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

14 - Fields Calculator Script Commands

Fields Calculator commands should be executed by the Field Overlays module, which is called FieldsReporter in HFSS 3D Layout scripts.

```
oModule = oDesign.GetModule("FieldsReporter")

oModule.CommandName <args>
```

The command associated with each of the following scripting commands will be a button pressed in the Fields Calculator.

AddNamedExpression

CalcOp

CalculatorRead

CalcStack

CalculatorWrite

ChangeGeomSettings

ClcEval

ClcMaterial

ClearAllNamedExpr

CopyNamedExprToStack

DeleteNamedExpr

EnterComplex

EnterComplexVector

EnterLine

EnterPoint

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnterQty

EnterScalar

EnterScalarFunc

EnterSurf

EnterVector

EnterVectorFunc

EnterVol

ExportOnGrid [Fields Calculator]

ExportToFile [Fields Calculator]

GetFieldsCalculatorExpressions

GetTopEntryValue

LoadNamedExpressions

SaveNamedExpressions

AddNamedExpression

Creates a named expression using the expression at the top of the stack.

<table>
<tr><td>UI Access</td><td>Click Add in the Fields Calculator window.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ExprName></td><td>String</td><td>Name for the new named expression.</td></tr>
<tr><td><FieldType></td><td>String</td><td>Type of field.</td></tr>
<tr><td><Expression></td><td>Array</td><td>Defines an expression to be solved in the calculator.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	AddNamedExpression(<*ExprName*>, <*FieldType*>)
Python Example	`oModule.AddNamedExpression("Mag_JxE", "Fields")`

```
oModule.AddNamedExpression("new2", "Fields",
```

CalcOp

Performs a calculator operation.

UI Access	Operation commands like **Mag**, **+**, etc.
Parameters	Name | Type | Description <*OpString*> | String | The text on the corresponding calculator button.
Return Value	None.

Python Syntax	CalcOp(<*OpString*>)
Python Example	`oModule.CalcOp("+")`

CalcRead(deprecated)

Reads a file that is written out by the CalcWrite command, and puts the result into a calculator numeric.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Read...** in the **Fields Calculator** dialog.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Name of the file to be read.</td></tr><tr><td><*SoluName*></td><td>String</td><td>Specified solution to read in.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variables to read.</td></tr></table>
Return Value	None.

Python Syntax	CalcRead(<*FileName*>, <*SoluName*>, <*VarArray*>)

Python Example	`oModule.CalcRead(`
	`  "c:\example.reg",`
	`  "Setup1: LastAdaptive",`
	`  ["Freq:=", "10GHz", "Phase:=", "0deg"])`

CalculatorRead

Gets a register file and applies it to the calculator stack.

UI Access	Click **Read...** in the **Fields Calculator** dialog.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Path to and including name of input register file.</td></tr><tr><td><*SoluName*></td><td>String</td><td>Specified solution to read in.</td></tr><tr><td><*FieldType*></td><td>String</td><td>Type of specified field.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable names, value pairs.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	CalculatorRead(<*FileName*>, <*SoluName*>, <*FieldType*>, <*VarArray*>)

Python Example	`oModule.CalculatorRead(`
	`  "c:\example.reg",`
	`  "Setup1: LastAdaptive", "Fields",`
	`  ["Freq:=", "10GHz", "Phase:=", "0deg"])`

CalcStack

Performs an operation on the stack.

UI Access	Stack operation buttons such as **Push** and **Pop**.		
Parameters	Name	Type	Description
	<*OpString*>	String	The text on the corresponding calculator button.
Return Value	None.		

Python Syntax	CalcStack(<*OpString*>)
Python Example	`oModule.CalcStack("push")`

CalculatorWrite

Writes contents of top register to file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Write...** in the **Fields Calculator** window.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*OutputFilePath*></td><td>String</td><td>Path to and including name of output register file.</td></tr><tr><td><*SoluNameArray*></td><td>Array</td><td>Array of strings containing solution names.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable names, value pairs.</td></tr></table>
Return Value	None

Python Syntax	CalculatorWrite(<*OutputFilePath*>, <*SoluNameArray*>, <*VarArray*>)

Python Example	`oModule.CalculatorWrite("C:\test.reg",`
	`  ["Solution:=", "Setup1 : LastAdaptive"],`
	`  ["Freq:=", "1GHz", "Phase:=", "0deg"]`
	`)`

ChangeGeomSettings

Changes the line discretization setting.

UI Access	In the **Fields Calculator** window, click on **Geom Settings...**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*LineDiscr*></td><td>Integer</td><td>Line discretization value.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ChangeGeomSettings(<*LineDiscr*>)
Python Example	`oModule.ChangeGeomSettings(500)`

ClcEval

Evaluates the expression at the top of the stack using the provided solution name and variable values.

UI Access	Click **Eval** in the **Fields Calculator** window.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluName*></td><td>String</td><td>Name of specified solution.</td></tr><tr><td><*VarArray*></td><td>Array</td><td>Array of variable name, value pairs.</td></tr><tr><td><*FieldType*></td><td>String</td><td>Optional. Type of specified field.</td></tr></table>
Return Value	None

Python Syntax	ClcEval(<*SoluName*>, <*VarArray*>, [Optional <*FieldType*>])

Python Example	`oModule.ClcEval("Setup1: LastAdaptive",`
	`  ["Freq:=", "10GHz",`
	`  "Phase:=", "0deg"])`

ClcMaterial

Performs a material operation on the top stack element.

UI Access	Click **Matl...** in the **Fields Calculator** window.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*MatString*></td><td>String</td><td>The material property to apply.</td></tr><tr><td><*OpString*></td><td>String</td><td>Name of operation. Possible values are "mult", or "div".</td></tr></table>
Return Value	None.

Python Syntax	ClcMaterial(<*MatString*>, <*OpString*>)
Python Example	`oModule.ClcMaterial("Permeability (mu)" "mult")`

ClcMaterialValue

Shows the value of the material property without performing any operation.

UI Access	Select **None** in the **Material Operation** dialog.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*MaterialName*></td><td>String</td><td>Name of specified material property.</td></tr></table>
Return Value	None.

Python Syntax	ClcMaterialValue(<*MaterialName*>)
Python Example	`oModule.ClcMaterialValue("Mass Density")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClearAllNamedExpr

Clears all user-defined named expressions from the list.

UI Access	Click **ClearAll** in the **Fields Calculator** window.
Parameters	None.
Return Value	None.

Python Syntax	ClearAllNamedExpr()
Python Example	`oModule.ClearAllNamedExpr()`

CopyNamedExprToStack

Copies the named expression selected to the calculator stack.

UI Access	Select a named expression and then click **Copy to stack**.
Parameters	Name | Type | Description <*ExprName*> | String | Name of the expression to be copied to the top of the stack.
Return Value	None.

Python Syntax	CopyNamedExprToStack(<*ExprName*>)
Python Example	`oModule.CopyNamedExprToStack("Mag_JxE")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteNamedExpr

Deletes the selected named expression from the list.

UI Access	Select a named expression and then click **Delete**.
Parameters	Name | Type | Description *<ExprName>* | String | Name of specified named expression.
Return Value	None.

Python Syntax	DeleteNamedExpr(<*ExprName*>)
Python Example	`oModule.DeleteNamedExpr("Mag_JxE")`

DoesNamedExpressionExists

Determines whether specified named expression exists.

UI Access	N/A
Parameters	Name | Type | Description *<ExpName>* | String | Name of the specified named expression.
Return Value	Boolean: • **1** - named expression exists.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• **0** - named expression does not exist.

Python Syntax	DoesNamedExpressionExists(<*ExpName*>)
Python Example	`oModule.DoesNamedExpressionExists("Mag_JxE")`

EnterComplex

Enters a complex number onto the stack.

UI Access	Click **Number**, and then click **Scalar**. **Complex** option is selected.
Parameters	Name | Type | Description <*ComplexNum*> | String | String of complex value. Ex. "1 + 2j".
Return Value	None.

Python Syntax	EnterComplex(<*ComplexNum*>)
Python Example	`oModule.EnterComplex("1 + 2 j")`

EnterComplexVector

Enters a complex vector onto the stack.

UI Access	Click **Number**, and then click **Vector**. **Complex** option is selected.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ComplexVector*></td><td>Array</td><td>Array of strings containing X, Y and Z complex values.</td></tr></table>
Return Value	None.

Python Syntax	EnterComplexVector(<*ComplexVector*>)

Python Example	`oModule.EnterComplexVector`
	`([`
	`  "1 + 2 j",`
	`  "1 + 2 j",`
	`  "1 + 2 j"`
	`])`

EnterCoord

Enters a coordinate system defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Coord**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*CoordName*></td><td>String</td><td>Name of a coordinate system defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnterCoord(<*CoordName*>)
Python Example	`oModule.EnterCoord("Global")`

EnterEdge

Enters an edge defined in the 3D Modeler editor.

UI Access	N/A
Parameters	Name | Type | Description <*EdgeName*> | String | Name of an edge defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterEdge(<*EdgeName*>)
Python Example	`oModule.EnterEdge("Edge_1")`

EnterLine

Enters a line defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Line**
Parameters	Name | Type | Description <*LineName*> | String | Name of a line defined in the 3D Modeler editor.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnterLine(<*LineName*>)
Python Example	`oModule.EnterLine("Line1")`

EnterOutputVar

Enters Output Vars, only valid for Eigenmode problems.

UI Access	Click **Geometry** and then select **Output Vars**.		
Parameters	Name	Type	Description
	<*VarName*>	String	Name of the output vars. Only 'freq' is supported.
	<*VarType*>	String	Type of the output vars. 'Complex' type.
Return Value	None.		

Python Syntax	EnterOutputVar(<*VarName*>, <*VarType*>)
Python Example	`oModule.EnterOutputVar("Freq", "Complex")`

EnterPoint

Enters a point defined in the 3D Modeler editor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Geometry** and then select **Point**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PointName*></td><td>String</td><td>Name of a point defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Python Syntax	EnterPoint(<*PointName*>)
Python Example	`oModule.EnterPoint("Point1")`

EnterQty

Enters a field quantity.

UI Access	Click **Quantity**, and then select from the list.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FieldQty*></td><td>String</td><td>The field quantity to be entered onto the stack.</td></tr></table>
Return Value	None.

Python Syntax	EnterQty(<*FieldQty*>)
Python Example	`oModule.EnterQty("E")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EnterScalar

Enters a scalar onto the stack.

UI Access	Click **Number** and then click **Scalar**. **Complex** option not selected.
Parameters	Name | Type | Description <*Scalar*> | Double | The real number to enter onto the stack.
Return Value	None.

Python Syntax	EnterScalar(<*Scalar*>)
Python Example	`oModule.EnterScalar(3.14159265358979)`

EnterScalarFunc

Enters a scalar function.

UI Access	Click **Function** and then select **Scalar**.
Parameters	Name | Type | Description <*VarName*> | String | Name of a variable to enter as a scalar function onto the stack.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnterScalarFunc(<*VarName*>)
Python Example	`oModule.EnterScalarFunc("Phase")`

EnterSurf

Enters a surface defined in the 3D Modeler editor.

UI Access	Click **Geometry** and then select **Surface**.
Parameters	Name | Type | Description <*SurfName*> | String | Name of a surface defined in the 3D Modeler editor.
Return Value	None.

Python Syntax	EnterSurf(<*SurfName*>)
Python Example	`oModule.EnterSurf("Rectangle1")`

EnterVector

Enters a vector onto the stack.

UI Access	Click **Number**, and then click **Vector**. **Complex** option not selected.
Parameters	Name | Type | Description <*VectorArray*> | Array | Array of strings containing X, Y and Z components of the vector.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EnterVector(<*VectorArray*>)
Python Example	`oModule.EnterVector([1.0, 1.0, 1.0])`

EnterVectorFunc

Enters a vector function.

UI Access	Click **Function** and then select **Vector**.
Parameters	Name | Type | Description <*VarNames*> | Array | Array of strings containing names of a variable for the X, Y, and Z coordinates, respectively, to enter as a vector function on the stack.
Return Value	None.

Python Syntax	EnterVectorFunc(<*VarNames*>)
Python Example	`oModuleEnterVectorFunc(["X", "Y", "Z"])`

EnterVol

Enters a volume defined in the 3D Modeler editor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Click **Geometry** and then select **Volume**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*VolumeName*></td><td>String</td><td>Name of a volume defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Python Syntax	EnterVol(<*VolumeName*>)
Python Example	`oModule.EnterVol("Box1")`

ExportOnGrid [Fields Calculator]

Evaluates the top stack element at a set of points specified by a grid and exports the data to a file.

UI Access	Click **Export**, and then click **On Grid**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OutputFile*></td><td>String</td><td>Name of the output file.</td></tr><tr><td><*Min*></td><td>Array</td><td>Minimum values for the coordinate components of the grid system</td></tr><tr><td><*Max*></td><td>Array</td><td>Maximum values for the coordinate components of the grid system</td></tr><tr><td><*Spacing*></td><td>Array</td><td>Spacing values for the coordinate components of the grid system</td></tr><tr><td><*SolnName*></td><td>String</td><td>Name of the simulation setup</td></tr><tr><td><*SolnParameters*></td><td>Array</td><td>Array of strings containing setup definitions.</td></tr><tr><td><*ExportOption*></td><td>Array</td><td>["NAME:ExportOption" <*IncludePointsInOutput*>: Boolean; specifies whether include points in the output file. <*RefCSName*>: String, specifies the reference coordinate system used by the model</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<*PtInSI*>: Boolean; if True, points are exported in SI. If False, points are exported in model units <*FieldInRefCS*>] :Boolean; if True, the points are exported in the reference coordinate system.
	<*IncludePoint-sInOutput*>	Boolean	Optional. Specifies whether include points in the output file.
	<*CSType*>	String	Optional. Type of coordinate system. "Cartesian" (default) | "Cylindrical" | "Spherical"
	<*Offsets*>	Array	Optional. Origin for the offset coordinate system. For Cartesian, x, y, z, for Cylindrical, R, Phi, Z, for Spherical, Rho, Theta, Phi.
	<*ByCount*>	Boolean	Optional.
Return Value	None.		

Note: Regarding the **ExportOnGrid** legacy script, which only has “IncludePtInOutput” argument, AEDT can still read it and assign other new arguments as default values. Those default values are RefCSName = “global”, PtInSI = “True”, FieldInRefCS = “False”

Python Syntax	ExportOnGrid(<*OutputFile*>, <*Min*>, <*Max*>, <*Spacing*>, <*SolnName*>, <*SolnParameters*>, <*ExportOption*>, "<*CSType*>", <*Offsets*>, <*ByCount*>)

Python Example	`oModule.ExportOnGrid("C:\offset_grid_model_unit_ref.fld",`
	`  ["-1mm", "16mm", "0mm"],`
	`  ["1mm", "18mm", "1mm"],`
	`  ["2mm", "2mm", "1mm"],`
	`  "4500MHz : LastAdaptive",`
	`  ["Freq:=", "4.5GHz","Phase:=" , "0deg"],`
	`  ["NAME:ExportOption",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "IncludePtInOutput:=" , True,
    "RefCSName:=" , "offset",
    "PtInSI:=" , False,
    "FieldInRefCS:=" , True
  ],
  "Cartesian", ["0mm", "0mm", "0mm"], False
  )
```

ExportToFile [Fields Calculator]

Evaluates the top stack element at a set of points specified in an external file and exports the data to a file.

> **Note:** Regarding to legacy **ExportToFile** script which only has “IncludePtInOutput” argument (the last Boolean one), AEDT can still read it and assign other new arguments as default values. Those default values are RefCSName = “global”, PtInSI = “True”, FieldInRefCS = “False”

UI Access	Click **Export**, and then click **To File**.

Parameters

Name	Type	Description
<*ReportName*>	String	Name of report to be exported.
<*FileName*>	String	Full path of the exported image file name; with extension of .txt - Post processor format file .csv - Comma-delimited data file .tab - Tab-separated file .dat - Ansys plot data file
<solution>		Solution Name
<freq>		frequency

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<phase> <Export Options>		phase Whether to include points in output, RefCSName, Pts in SI units <boolean>, Field in RefCS, <boolean>.
Return Value	None.		

Python Syntax	ExportToFile(<*ReportName*>, <*FileName*>, <solution>, ["Freq:=", <*freq*>, "Phase:=", "<*phase*>], <ExportOptions>)

Python Example	`oModule.ExportToFile("C:\\offset_file_model_unit_ref.fld", "C:\\offset_SI.pts",`
	`  "4500MHz : LastAdaptive",`
	`  [`
	`  "Freq:=" , "4.5GHz",`
	`  "Phase:=" , "0deg"`
	`  ],`
	`  [`
	`  "NAME:ExportOption",`
	`  "IncludePtInOutput:=" , True,`
	`  "RefCSName:=" , "offset",`
	`  "PtInSI:=" , False,`
	`  "FieldInRefCS:=" , True`
	`  ])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetFieldsCalculatorExpressions

Returns a list of named expressions available in the Fields Calculator window. The list will vary according to the active design.

<table>
<tr><td>UI Access</td><td>In the Project Manager tree, right click Field Overlays, and select Calculator</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><fieldType></td><td>String</td><td>Optional; the available options will be determined by the active design. For all designs, the default is "<code>Fields</code>".</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>A list of strings containing named expressions for the specified fieldType. If the user does not provide a fieldType, all named expressions are returned, including user-defined expressions. Each entry in the list follows the format: "Name = expression".</td></tr>
</table>

Python Syntax	GetFieldsCalculatorExpressions(*<fieldType>*)

Python Example	`oModule = oDesign.GetModule("FieldsReporter")`
	
	`namedExpressions = oModule.GetFieldsCalculatorExpressions()`
	
	Example of return value for a Maxwell Transient design when fieldType is **"Fields"**:
	
	`['Mag_H = Mag(Smooth(<Hx,Hy,Hz>))', 'Mag_B = Mag(Smooth(<Bx,By,Bz>))', 'Mag_J = Mag(Smooth(<Jx,Jy,Jz>))', 'H_Vector = Smooth(<Hx,Hy,Hz>)', 'B_Vector = Smooth (<Bx,By,Bz>)', 'J_Vector = Smooth(<Jx,Jy,Jz>)', 'energy = Smooth(Energy)', 'coEn-ergy = Smooth(CoEnergy)', 'appEnergy = Smooth(App_Energy)', 'Ohmic_Loss = Smooth (Ohmic-Loss)', 'Total_Loss = Smooth(TotalLoss)', 'Temperature = Smooth(Temp)', 'Volume_Force_Density = <VolumeFor-ceDensityx,VolumeForceDensityy,VolumeForceDensityz>', 'Surface_Force_Density = <SurfaceForceDensityx,SurfaceForceDensityy,SurfaceForceDensityz>', 'Demag_Coef = Smooth(DemagCoef)', 'Surface_Loss_Density = SurfaceLossDensity']`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetTopEntryValue

Gets the value of the top entry of the calculator stack.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SolnName*></td><td>String</td><td>Name of specified solution.</td></tr><tr><td><*VarVals*></td><td>Array</td><td>Array of variable name, value pairs.</td></tr></table>
Return Value	Array of strings containing top entry values.

Python Syntax	GetTopEntryValue(<*SolnName*>, <*VarVals*>)

Python Example	`oModule.GetTopEntryValue(`
	`  "Setup1:LastAdaptive",`
	`  ["Freq:=", "1GHz", "Phase:=", "0deg", "x_size:=", "2mm"]`
	`)`

LoadNamedExpressions

Loads a named expression definition from a saved file.

UI Access	In the Fields Calculator, click **Load From...** in the Library area.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Filename and full path to the file to hold the named expression definition.</td></tr><tr><td><*FieldType*></td><td>String</td><td>For products with just one filed type, it is set to "Fields".</td></tr><tr><td><*NamedExpr*></td><td>Array</td><td>rray of strings containing the names of expression definitions to load from the file.</td></tr></table>
Return Value	None.

Python Syntax	LoadNamedExpressions(<*FileName*>, <*FieldType*>, <*NamedExpr*>)

Python Example	`oModule.LoadNamedExpressions(`
	`  "C:\Ansoft\PersonalLib\smth.clc",`
	`  "Fields", ["smoothedtemp"])`

SaveNamedExpressions

Saves a named expression definition to a file.

UI Access	In the Fields Calculator, click **Save To...** in the Library area.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Filename and full path to the file to hold the named expression definition.</td></tr><tr><td><NamedExprs></td><td>Array</td><td>Array of strings containing the names of expression definitions to load from the file.</td></tr><tr><td><*Overwrite*></td><td>Boolean</td><td>Specifies whether to overwrite the file.</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SaveNamedExpressions(<*FileName*>, <*NamedExprs*>, <*Overwrite*>)

Python Example	`oModule.SaveNamedExpressions(`
	`"C:\Ansoft\PersonalLib\smth.clc",`
	`["smoothedtemp"], True)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

15 - Radiation Module Script Commands

Radiation field commands should be executed by the **RadField** module.

oModule = oDesign.GetModule("RadField")

oModule.CommandName <args>

Conventions Used in this Chapter

`<SetupName>`

Type: <string>

Name of a radiation setup.

`<FaceListName>`

Type: <string>

Name of a qualifying face list. Used for specifying custom radiation surfaces. In order to be valid for use in a radiation surface, the face list should not contain any faces on PML objects and should contain only model faces.

`<CSName>`

Type: string

Name of a coordinate system.

The topics for this section include:

General Commands Recognized by the Radiation Module

Script Commands for Creating and Modifying Radiation Setups

Script Commands for Modifying Antenna Array Setups

Script Commands for Exporting Antenna Parameters and Max Field Parameters

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

General Commands Recognized by the Radiation Module

DeleteFarFieldSetup

DeleteNearFieldSetup

GetSetupNames

RenameSetup

SetPropValue [Radiation]

CopyRadFieldSetup

Copies a radiation field setup to clipboard.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of radiation field setup to be copied.
Return Value	None.

Python Syntax	CopyRadFieldSetup(<*SetupName*>)
Python Example	`oModule.CopyRadFieldSetup("RadField1")`

DeleteFarFieldSetup

Deletes an existing far-field setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click on a far field setup under **Radiation** in the project tree, select **Delete**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>Array</td><td>Name of setup(s) to be deleted.</td></tr></table>
Return Value	None.

Python Syntax	DeleteFarFieldSetup(<*SetupName*>)
Python Example	`oModule.DeleteFarFieldSetup(["Infinite Sphere1"])`

DeleteNearFieldSetup

Deletes an existing near-field line radiation setup.

UI Access	**Project Manager** > **Radiation**. Right-click a setup and click **Delete**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NameArray*></td><td>Array</td><td>List of fields for deletion.</td></tr></table>
Return Value	None.

Python Syntax	DeleteNearFieldSetup(*NameArray*)
Python Example	`oModule.DeleteNearFieldSetup(['Line1', 'Sphere1', 'Box2', 'Rectangle1'])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteSetup

Deletes specified radiation field setup(s).

UI Access	Right-click on a setup under **Radiation** in the project, select **Delete**.
Parameters	Name | Type | Description <*SetupName*> | Array | Array of setup names to be deleted.
Return Value	None.

Python Syntax	DeleteSetup(<*SetupName*>)
Python Example	`oModule.DeleteSetup(["Rad Field1"])`

EditRadiatedPowerCalculationMethod

Selects the method used to calculate radiated power.

UI Access	Right-click Radiation in the project tree, select **Radiated Power Calculation Method**.
Parameters	Name | Type | Description <*PowerCalculationMethod*> | String | One of: "Auto", "Raidation Surface Integral", or "Far Field Integral".
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditRadiatedPowerCalculationMethod(<*PowerCalculationMethod*>)

Python Example	`oModule.EditRadiatedPowerCalculationMethod("Auto")`
	`oModule.EditRadiatedPowerCalculationMethod("Radiation Surface Integral")`
	`oModule.EditRadiatedPowerCalculationMethod("Far Field Integral")`

GetChildNames [Radiation]

Use: Gets setupup names .

Syntax: GetChildNames()

Return Value: All setup names.

Parameters: Optional, this parameter is ignored.

Python Syntax	GetChildNames ()
Python Example	`GetChildNames()  //return all setup names.`

GetChildObject [Radiation]

Gets a Setup Object of the Radiation module.

Python Syntax	GetChildObject()
Python Example	`oOverlay= oRadModule.GetChildObject('Antenna Parameter Overlay1')` `oSphere = oRadModule.GetChildObject('Infinite Sphere1')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetChildTypes [Radiation]

Use: Gets child types of queried Radiation module object.

Syntax: GetChildTypes()

Return Value: Returns empty array.

Python Syntax	GetChildTypes ()
Python Example	`oRad = oDesign.GetChildObject("RadField")` `oRadModule.GetChildTypes()`

GetPropNames [Radiation]

Use: Radiation module does not have its own property, this function always returns empty array.

Syntax: GetPropNames(bIncludeReadOnly)

Return Value: Returns empty set.

Parameters: bIncludeReadOnly—optional, default to True.

Python Syntax	GetPropNames ()

Python Example	`oRadModule.GetPropNames()`
	`oRadModule.GetPropNames(True)`
	`oRadModule.GetPropNames(False)`

GetPropValue [Radiation]

Get the property value for a setup object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	Name | Type | Description PropPath Value | text string | A Property path begin with a setup name. See property path discussion here.
Return Value	The property value.

Python Syntax	GetPropValue(propPath)
Python Example	`oRadModule.GetPropValue('Line1/Num Points') //Get the Line1 setups' "Num Points" property value.`

GetSetupNames

Gets the names of far field and near field radiation setups in a design.

UI Access	N/A
Parameters	Name | Type | Description <*RadiationType*> | String | Type of radiation.
Return Value	Array of strings containing setup names.

Python Syntax	GetSetupNames(<*RadiationType*>)
Python Example	`oModule.GetSetupNames("Infinite Sphere")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PasteRadFieldSetup

Pastes a copied radiation field setup.

UI Access	N/A
Parameters	None.
Return Value	String name of pasted setup.

Python Syntax	PasteRadFieldSetup()
Python Example	`oModule.PasteRadFieldSetup()`

RenameSetup [Radiation]

Renames an existing radiation setup.

UI Access	Right-click a radiation setup in the project tree, and then click **Rename** on the shortcut menu.		
Parameters	Name	Type	Description
	<*OldName*>	String	Name of specified setup to be renamed.
	<*NewName*>	String	New name for the setup.
Return Value	None.		

Python Syntax	RenameSetup(<*OldName*>, <*NewName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.RenameSetup("Sphere1", "MyNearSphere")`

SetPropValue [Radiation]

Sets the property value for the active Radiation property object.

UI Access	Edit Properties on ProjectTree objects
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>propPath</td><td>text string</td><td>A Property path begin with a setup name. See property path discussion here.</td></tr><tr><td>Value</td><td>Text String, Number, or Boolean</td><td>New value data type is depending on the property type.</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Python Syntax	SetPropValue(propPath, value)
Python Example	`oRadModule.SetPropValue('Line1/Num Points', 100) ; Set the Line1 setups' "Num Points" property to 100.`

Script Commands for Creating and Modifying Radiation Setups

EditBoxSetup

EditFarFieldSphereSetup

EditNearFieldLineSetup

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditNearFieldRectangleSetup

EditNearFieldSphereSetup

InsertBoxSetup

InsertFarFieldSphereSetup

InsertNearFieldLineSetup

InsertNearFieldRectangleSetup

InsertNearFieldSphereSetup

AddAntennaOverlay

Saves a table of antenna parameters for use an overlay on far field plots.

UI Access	Click on **Save for Overlay** in the **Antenna Parameters** dialog.		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<AntennaParams>	Array	Structured array. ["NAME:<AntennaOverlayName>", "Setup:=", <string far field radiation setup name>, "Solution:=", <string for the Solution Setup and Pass>, "IntrinsicVariation:=", <string for variation definition>, "UseNominalDesign:=", <boolean>, "DesignVariation:=", <string>, "Calculations:=" , <list of each antenna parameter to display>]
Return Value	None.		

Python Syntax	AddAntennaOverlay(*<AntennaParams>*)
Python Example	oModule.AddAntennaOverlay(["NAME:Antenna Parameter Overlay1", "Setup:=", "Infinite Sphere1", "Solution:=", "4500MHz : LastAdaptive", "IntrinsicVariation:=", "Freq=4.5GHz ", "UseNominalDesign:=", False, "DesignVariation:=", "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Calculations:=", "Max U,Peak Directivity,Peak Gain,Radiation Efficiency, Front To Back Ratio,Total Efficiency,System Efficiency"])

AddRadFieldSourceGroup

Adds source group definitions for radiation field setup.

UI Access	Right-click on **Radiation** in the project tree, select **Insert Far Field Setup** > **Source Group...**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SourceGroupParams*></td><td>Array</td><td>Structured array ["NAME:<SourceGroupName>", ["NAME:Sources", <SourceName>, <SourceName>...]]</td></tr></table>
Return Value	None.

Python Syntax	AddRadFieldSourceGroup(<*SourceGroupParams*>)
Python Example	oModule.AddRadFieldSourceGroup(["NAME:Source Group1", ["NAME:Sources", "feedpin1_T1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"feedpin1_T2"]])

EditAntennaOverlay

Edits parameters in an existing antenna overlay.

UI Access	N/A
Parameters	See table below
Return Value	None.

Name	Type	Description
<AntennaOverlayName>	String	Name of the antenna overlay to be edited.
<AntennaParams>	Array	Structured array. ["NAME:<AntennaOverlayName>", "Setup:=", <string far field radiation setup name>, "Solution:=", <string for the Solution Setup and Pass>, "IntrinsicVariation:=", <string for variation definition>, "UseNominalDesign:=", <boolean>, "DesignVariation:=", <string>, "Calculations:=" , <list of each antenna parameter to display>]

Python Syntax	EditAntennaOverlay(*<AntennaOverlayName>*, *<AntennaParams>*)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oModule.EditAntennaOverlay("Antenna Parameter Overlay1", ["NAME:Antenna Parameter Overlay1", "Setup:=", "Infinite Sphere1", "Solution:=", "4500MHz : LastAdaptive", "IntrinsicVariation:=", "Freq=4.5GHz ", "UseNominalDesign:=", False, "DesignVariation:=", "", "Calculations:=", "Max U,Peak Directivity,Peak Gain,Radiation Efficiency, Front To Back Ratio,Total Efficiency,System Efficiency"])

EditBoxSetup

Edits a near-field box radiation setup.

UI Access	**Project Manager** > **Radiation** > **Box**. Double-click and edit information.
Parameters	Name: *<BoxParams>*; Type: Array; Description: Structured array. ["NAME:<SetupName>", "UseCustomRadiationSurface:=", <bool>, "CustomRadiationSurface:=", <FaceListName>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Length:=", <string>, "Width:=", <string>, "LengthSamples:=", <int>, "WidthSamples:=", <int>, "CoordSystem:=", <CSname> "Height:=", <string>, "HeightSamples:=", <int>] For UseCustomRadiationSurface parameter, provide value if True. If False, radiation boundary/PML boundaries will be used as radiation surfaces.
Return Value	None.		

Python Syntax	EditBoxSetup(<*BoxParams*>)
Python Example	oModule.EditBoxSetup(['NAME:MyBox', 'UseCustomRadiationSurface:=', False, 'Length:=', '20mm', 'Width:=', '20mm', 'LengthSamples:=', 21, 'WidthSamples:=', 21, 'CoordSystem:=', 'Global', 'Height:=', '20mm',

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	'HeightSamples:=', 21])

EditFarFieldSphereSetup

Modifies an existing far-field sphere setup.

UI Access	Double-click a radiation setup in the project tree to modify its settings.
Parameters	See table below

Name	Type	Description
<FarFieldSphereName>	String	Name of the Far Field Sphere Radiation Setup to be edited.
<FarFieldSphereParams>	Array	Structured array. ["NAME:<InfSphereName>", "UseCustomRadiationSurface:=", <boolean>, "CSDefinition:=", <string, "Theta-Phi" | "Az Over EI" | "EI Over AS">, "Polarization:=", <string, "Linear" | "Slant">, "ThetaStart:=", "<value><unit>", "ThetaStop:=", "<value><unit>", "ThetaStep:=", "<value><unit>", "PhiStart:=", "<value><unit>", "PhiStop:=", "<value><unit>", "PhiStep:=", "<value><unit>",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"UseLocalCS:=", boolean]
Return Value	None

Python Syntax	EditFarFieldSphereSetup(<*FarFieldSphereName*>, <*FarFieldSphereParams*>)
Python Example	oModule.EditFarFieldSphereSetup("Infinite Sphere1", ["NAME:Infinite Sphere1", "UseCustomRadiationSurface:=", False, "ThetaStart:=" , "0deg", "ThetaStop:=" , "180deg", "ThetaStep:=" , "3deg", "PhiStart:=" , "-180deg", "PhiStop:=" , "180deg", "PhiStep:=" , "3deg", "UseLocalCS:=" , False])

EditInfiniteSphereSetup

Modifies an infinite sphere radiation field setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Double-click on a setup under Radiation in the project tree.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><InfSphereName></td><td>String</td><td>Name of the infinite sphere setup to be edited.</td></tr>
<tr><td><InfSphereParams></td><td>Array</td><td>Structured array.
<pre>["NAME:<InfSphereName>",
  "UseCustomRadiationSurface:=", <boolean>,
  "CSDefinition:=", <string, "Theta-Phi" | "Az Over EI" | "EI Over AS">,
  "Polarization:=", <string, "Linear" | "Slant">,
  "ThetaStart:=", "<value><unit>",
  "ThetaStop:=", "<value><unit>",
  "ThetaStep:=", "<value><unit>",
  "PhiStart:=", "<value><unit>",
  "PhiStop:=", "<value><unit>",
  "PhiStep:=", "<value><unit>",
  "UseLocalCS:=", boolean]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	EditInfiniteSphereSetup(<*InfSphereName*>, <*InfSphereParams*>)
Python Example	`oModule.EditInfiniteSphereSetup("Infinite Sphere1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:Infinite Sphere1",
  "UseCustomRadiationSurface:=", False,
  "CSDefinition:="        , "Theta-Phi",
  "Polarization:="        , "Linear",
  "ThetaStart:="          , "0deg",
  "ThetaStop:="           , "180deg",
  "ThetaStep:="           , "2deg",
  "PhiStart:="            , "-180deg",
  "PhiStop:="             , "180deg",
  "PhiStep:="             , "2deg",
  "UseLocalCS:="          , False
])
```

EditLineSetup

Modifies a near field line setup.

UI Access	Double-click on a setup under **Radiation** in the project tree.
Parameters	See below

Name	Type	Description
<*SetupName*>	String	Name of the setup to be edited.
<*LineParams*>	Array	Structured array. `["NAME:<LineName>",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"UseCustomRadiationSurface:=", <boolean>, "Line:=", <string, name of specified line>, "NumPts:=", "<integer, number of points>"]
Return Value	None.

Python Syntax	EditLineSetup(<*SetupName*>, <*LineParams*>)
Python Example	oModule.EditLineSetup("Line1", ["NAME:Line1", "UseCustomRadiationSurface:=", False, "Line:=" , "Polyline1", "NumPts:=" , "1000"])

EditNearFieldBoxSetup

Modifies a near field box radiation field setup.

UI Access	N/A
Parameters	Name | Type | Description <*BoxSetupName*> | String | Name of the near field box setup to be edited.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<BoxSetupParams>	Array	Structured array. ["NAME:<BoxSetupName>", "UseCustomRadiationSurface:=", <boolean>, "Length:=", "<value><unit>", "Width:=", "<value><unit>", "LengthSamples:=", <value>, "WidthSamples:=", <value>, "CoordSystem:=", <string name of coordinate system>, "Height:=", "<value><unit>", "HeightSamples:=", <value>]
Return Value	None.		

Python Syntax	EditNearFieldBoxSetup(*<BoxSetupName>*, *<BoxSetupParams>*)
Python Example	oModule.EditNearFieldBoxSetup("Box1", ["NAME:Box1", "UseCustomRadiationSurface:=", False, "Length:=" , "20meter", "Width:=" , "20meter", "LengthSamples:=" , 21,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"WidthSamples:=" , 21, "CoordSystem:=" , "Global", "Height:=" , "20meter", "HeightSamples:=" , 21])

EditNearFieldLineSetup

Modifies a near field line setup.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SetupName*></td><td>String</td><td>Name of the setup to be edited.</td></tr><tr><td><*LineParams*></td><td>Array</td><td>Structured array. ["NAME:<LineName>", "UseCustomRadiationSurface:=", <boolean>, "Line:=", <string, name of specified line>, "NumPts:=", "<integer, number of points>"]</td></tr></table>
Return Value	None.

Python Syntax	EditNearFieldLineSetup(<*SetupName*>, <*LineParams*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.EditNearFieldLineSetup("Line1",`
	`[`
	`  "NAME:Line1",`
	`  "UseCustomRadiationSurface:=", False,`
	`  "Line:="  , "Polyline1",`
	`  "NumPts:="  , "1000"`
	`])`

EditNearFieldRectangleSetup

Modifies a near field rectangle radiation setup.

UI Access	N/A.
Parameters	

Name	Type	Description
<RectangleSetupName>	String	Name of rectangle radiation setup to be edited.

<RectangleSetupParams>	Array	Structured array.
		`["NAME:<RectangleSetupName>",`
		`  "UseCustomRadiationSurface:=", <boolean>,`
		`  "Length:=", "<value><unit>",`
		`  "Width:=", "<value><unit>",`
		`  "LengthSamples:=", <value>,`
		`  "WidthSamples:=", <value>,`
		`  "CoordSystem:=", <string name of coordinate sys-`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	tem>]
Return Value	None.

Python Syntax	EditNearFieldRectangleSetup(<*RectangleSetupName*>, <*RectangleSetupParams*>)
Python Example	oModule.EditNearFieldRectangleSetup("Rectangle1", ["NAME:Rectangle1", "UseCustomRadiationSurface:=", False, "Length:=" , "20meter", "Width:=" , "20meter", "LengthSamples:=" , 41, "WidthSamples:=" , 41, "CoordSystem:=" , "Global"])

EditNearFieldSphereSetup

Modifies a near field sphere radiation setup.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*SphereSetupName*>	String	Name of sphere radiation field setup to be edited.

<*SphereSetupParams*>	Array	Structured array.
		`["NAME:<SphereSetupName>",`
		`"UseCustomRadiationSurface:=", <boolean>,`
		`"Radius:=", "<value><unit>",`
		`"ThetaStart:=", "<value><unit>",`
		`"ThetaStop:=", "<value><unit>",`
		`"ThetaStep:=", "<value><unit>",`
		`"PhiStart:=", "<value><unit>",`
		`"PhiStop:=", "<value><unit>",`
		`"PhiStep:=", "<value><unit>",`
		`"UseLocalCS:=", <boolean>]`

Return Value

None.

Python Syntax	EditNearFieldSphereSetup(<*SphereSetupName*>, <*SphereSetupParams*>)

Python Example	`oModule.EditNearFieldSphereSetup("Sphere1",`
	`[`
	`"NAME:Sphere2",`
	`"UseCustomRadiationSurface:=", False,`
	`"Radius:="  , "20meter",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "ThetaStart:="          , "0deg",
  "ThetaStop:="           , "180deg",
  "ThetaStep:="           , "2deg",
  "PhiStart:="            , "-180deg",
  "PhiStop:="             , "180deg",
  "PhiStep:="             , "2deg",
  "UseLocalCS:="          , False
])
```

EditRadFieldSourceGroup

Modifies an existing source group definitions for radiation field setup.

UI Access	Double-click on a source group setup under **Radiation** in the project tree.
Parameters	See table below
Return Value	None.

Name	Type	Description
<*SourceGroupName*>	String	Name of the source group setup to be edited.

<*SourceGroupParams*>	Array	Structured array
		`["NAME:<SourceGroupName>",`
		`  ["NAME:Sources",`
		`  <SourceName>, <SourceName>...]]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditRadFieldSourceGroup(<*SourceGroupName*>, <*SourceGroupParams*>)

Python Example	`oModule.EditRadFieldSourceGroup("Source Group1",`
	`[`
	`  "NAME:Source Group1",`
	`  ["NAME:Sources",`
	`  "IncPWave1",`
	`  "IncPWave2"]`
	`])`

EditRectangleSetup

Modifies a near field rectangle radiation setup.

UI Access	Double-click on a rectangle setup under **Radiation** in the project tree.

Parameters

Name	Type	Description
<*RectangleSetupName*>	String	Name of rectangle radiation setup to be edited.

<*RectangleSetupParams*>	Array	Structured array.
		`["NAME:<RectangleSetupName>",`
		`  "UseCustomRadiationSurface:=", <boolean>,`
		`  "Length:=", "<value><unit>",`
		`  "Width:=", "<value><unit>",`
		`  "LengthSamples:=", <value>,`
		`  "WidthSamples:=", <value>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"CoordSystem:=", <string name of coordinate system>]
Return Value	None.		

Python Syntax	EditRectangleSetup(<*RectangleSetupName*>, <*RectangleSetupParams*>)
Python Example	oModule.EditRectangleSetup("Rectangle1", ["NAME:Rectangle1", "UseCustomRadiationSurface:=", False, "Length:=" , "20meter", "Width:=" , "20meter", "LengthSamples:=" , 41, "WidthSamples:=" , 41, "CoordSystem:=" , "Global"])

EditSphereSetup

Modifies a near field sphere radiation setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Double-click on a sphere setup under Radiation in the project tree.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><*SphereSetupName*></td><td>String</td><td>Name of sphere radiation field setup to be edited.</td></tr>
<tr><td><*SphereSetupParams*></td><td>Array</td><td>Structured array.
["NAME:<SphereSetupName>",
"UseCustomRadiationSurface:=", <boolean>,
"Radius:=", "<value><unit>",
"ThetaStart:=", "<value><unit>",
"ThetaStop:=", "<value><unit>",
"ThetaStep:=", "<value><unit>",
"PhiStart:=", "<value><unit>",
"PhiStop:=", "<value><unit>",
"PhiStep:=", "<value><unit>",
"UseLocalCS:=", <boolean>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	EditSphereSetup(<*SphereSetupName*>, <*SphereSetupParams*>)
Python Example	oModule.EditSphereSetup("Sphere1", ["NAME:Sphere2", "UseCustomRadiationSurface:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "Radius:="              , "20meter",
   "ThetaStart:="          , "0deg",
   "ThetaStop:="           , "180deg",
   "ThetaStep:="           , "2deg",
   "PhiStart:="            , "-180deg",
   "PhiStop:="             , "180deg",
   "PhiStep:="             , "2deg",
   "UseLocalCS:="          , False
])
```

InsertBoxSetup

Inserts a near-field box radiation setup.

UI Access	**HFSS** > **Radiation** > **Insert Near Field Setup** > **Box**.

Parameters

Name	Type	Description

<BoxParams>	Array	Structured array.
		`["NAME:<SetupName>",`
		`"UseCustomRadiationSurface:=", <bool>,`
		`"CustomRadiationSurface:=", <FaceListName>,`
		`"Length:=", <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Width:=", <string>, "LengthSamples:=", <int>, "WidthSamples:=", <int>, "CoordSystem:=", <CSname> "Height:=", <string>, "HeightSamples:=", <int>] For UseCustomRadiationSurface parameter, provide value if True. If False, radiation boundary/PML boundaries will be used as radiation surfaces.
Return Value	None.		

Python Syntax	InsertBoxSetup(<*BoxParams*>)
Python Example	oModule.InsertBoxSetup(['NAME:MyBox', 'UseCustomRadiationSurface:=', False, 'Length:=', '20mm', 'Width:=', '20mm', 'LengthSamples:=', 21, 'WidthSamples:=', 21, 'CoordSystem:=', 'Global', 'Height:=', '20mm', 'HeightSamples:=', 21

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

InsertFarFieldSphereSetup

Inserts a far field infinite sphere setup.

UI Access	N/A
Parameters	See table below
Return Value	None.

Name	Type	Description

<*SphereParams*>	Array	Structured array.
		`["NAME:<InfSphereName>",`
		`"UseCustomRadiationSurface:=", <boolean>,`
		`"CSDefinition:=", <string, "Theta-Phi" | "Az Over EI" | "EI Over AS">,`
		`"Polarization:=", <string, "Linear" | "Slant">,`
		`"ThetaStart:=", "<value><unit>",`
		`"ThetaStop:=", "<value><unit>",`
		`"ThetaStep:=", "<value><unit>",`
		`"PhiStart:=", "<value><unit>",`
		`"PhiStop:=", "<value><unit>",`
		`"PhiStep:=", "<value><unit>",`
		`"UseLocalCS:=", boolean]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	InsertFarFieldSphereSetup(<*SphereParams*>)
Python Example	oModule.InsertFarFieldSphereSetup(["NAME:Infinite Sphere1", "UseCustomRadiationSurface:=", False, "CSDefinition:=" , "Theta-Phi", "Polarization:=" , "Linear", "ThetaStart:=" , "0deg", "ThetaStop:=" , "180deg", "ThetaStep:=" , "2deg", "PhiStart:=" , "-180deg", "PhiStop:=" , "180deg", "PhiStep:=" , "2deg", "UseLocalCS:=" , False])

InsertInfiniteSphereSetup

Inserts an infinite sphere radiation field setup.

UI Access	Right-click on **Radiation** in the project tree, select **Insert Far Field Setup** > **Infinite Sphere...**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description

<*InfSphereParams*>	Array	Structured array.
		`["NAME:<InfSphereName>",`
		`"UseCustomRadiationSurface:=", <boolean>,`
		`"CSDefinition:=", <string, "Theta-Phi" | "Az Over EI" | "EI Over AS">,`
		`"Polarization:=", <string, "Linear" | "Slant">,`
		`"ThetaStart:=", "<value><unit>",`
		`"ThetaStop:=", "<value><unit>",`
		`"ThetaStep:=", "<value><unit>",`
		`"PhiStart:=", "<value><unit>",`
		`"PhiStop:=", "<value><unit>",`
		`"PhiStep:=", "<value><unit>",`
		`"UseLocalCS:=", boolean]`

Return Value

None.

Python Syntax	InsertInfiniteSphereSetup(<*InfSphereParams*>)

Python Example	`oModule.InsertInfiniteSphereSetup(`
	`[`
	`"NAME:Infinite Sphere1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "UseCustomRadiationSurface:=", False,
  "CSDefinition:="        , "Theta-Phi",
  "Polarization:="        , "Linear",
  "ThetaStart:="          , "0deg",
  "ThetaStop:="           , "180deg",
  "ThetaStep:="           , "2deg",
  "PhiStart:="            , "-180deg",
  "PhiStop:="             , "180deg",
  "PhiStep:="             , "2deg",
  "UseLocalCS:="          , False
])
```

InsertLineSetup

Inserts a near field line setup.

UI Access	Right-click on **Radiation** in the project tree, select **Insert Near Field Setup** > **Line...**
Parameters	

Name	Type	Description

<LineParams>	Array	Structured array.
		`["NAME:<LineName>",`
		`  "UseCustomRadiationSurface:=", <boolean>,`
		`  "Line:=", <string, name of specified line>,`
		`  "NumPts:=", "<integer, number of points>"]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	InsertLineSetup(<*LineParams*>)

Python Example	`oModule.InsertLineSetup(`
	`[`
	`  "NAME:Line1",`
	`  "UseCustomRadiationSurface:=", False,`
	`  "Line:="  , "Polyline1",`
	`  "NumPts:="  , "1000"`
	`])`

InsertNearFieldBoxSetup

Inserts a near field box radiation field setup.

UI Access	N/A

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*BoxSetupParams*></td><td>Array</td><td>Structured array.
	`["NAME:<BoxSetupName>",`
	`  "UseCustomRadiationSurface:=", <boolean>,`
	`  "Length:=", "<value><unit>",`</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Width:=", "<value><unit>", "LengthSamples:=", <value>, "WidthSamples:=", <value>, "CoordSystem:=", <string name of coordinate system>, "Height:=", "<value><unit>", "HeightSamples:=", <value>]
Return Value	None.		

Python Syntax	InsertNearFieldBoxSetup(<*BoxSetupParams*>)
Python Example	oModule.InsertNearFieldBoxSetup(["NAME:Box1", "UseCustomRadiationSurface:=", False, "Length:=" , "20meter", "Width:=" , "20meter", "LengthSamples:=" , 21, "WidthSamples:=" , 21, "CoordSystem:=" , "Global", "Height:=" , "20meter", "HeightSamples:=" , 21

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

InsertNearFieldLineSetup

Inserts a near field line setup.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*LineParams*></td><td>Array</td><td>Structured array. ["NAME:<LineName>", "UseCustomRadiationSurface:=", <boolean>, "Line:=", <string, name of specified line>, "NumPts:=", "<integer, number of points>"]</td></tr></table>
Return Value	None.

Python Syntax	InsertNearFieldLineSetup(<*LineParams*>)
Python Example	oModule.InsertNearFieldLineSetup(["NAME:Line1", "UseCustomRadiationSurface:=", False, "Line:=" , "Polyline1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NumPts:=" , "1000"])

InsertNearFieldRectangleSetup

Inserts a near field rectangle radiation setup.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*RectangleSetupParams*></td><td>Array</td><td>Structured array. ["NAME:<RectangleSetupName>", "UseCustomRadiationSurface:=", <boolean>, "Length:=", "<value><unit>", "Width:=", "<value><unit>", "LengthSamples:=", <value>, "WidthSamples:=", <value>, "CoordSystem:=", <string name of coordinate system>]</td></tr></table>
Return Value	None.

Python Syntax	InsertNearFieldRectangleSetup(<*RectangleSetupParams*>)
Python Example	oModule.InsertNearFieldRectangleSetup(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:Rectangle1",
  "UseCustomRadiationSurface:=", False,
  "Length:="              , "20meter",
  "Width:="               , "20meter",
  "LengthSamples:="       , 41,
  "WidthSamples:="        , 41,
  "CoordSystem:="         , "Global"
])
```

InsertNearFieldSphereSetup

Inserts a near field sphere radiation setup.

UI Access	N/A
Parameters	

Name	Type	Description

<*SphereSetupParams*>	Array	Structured array.
		`["NAME:<SphereSetupName>",`
		`  "UseCustomRadiationSurface:=", <boolean>,`
		`  "Radius:=", "<value><unit>",`
		`  "ThetaStart:=", "<value><unit>",`
		`  "ThetaStop:=", "<value><unit>",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"ThetaStep:=", "<value><unit>", "PhiStart:=", "<value><unit>", "PhiStop:=", "<value><unit>", "PhiStep:=", "<value><unit>", "UseLocalCS:=", <boolean>]
Return Value	None.		

Python Syntax	InsertNearFieldSphereSetup(<*SphereSetupParams*>)
Python Example	oModule.InsertNearFieldSphereSetup(["NAME:Sphere2", "UseCustomRadiationSurface:=", False, "Radius:=" , "20meter", "ThetaStart:=" , "0deg", "ThetaStop:=" , "180deg", "ThetaStep:=" , "2deg", "PhiStart:=" , "-180deg", "PhiStop:=" , "180deg", "PhiStep:=" , "2deg", "UseLocalCS:=" , False

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

InsertRectangleSetup

Inserts a near field rectangle radiation setup.

UI Access	Right-click on **Radiation** in the project tree ,select **Insert Near Field Setup** > **Rectangle...**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*RectangleSetupParams*></td><td>Array</td><td>Structured array. ["NAME:<RectangleSetupName>", "UseCustomRadiationSurface:=", <boolean>, "Length:=", "<value><unit>", "Width:=", "<value><unit>", "LengthSamples:=", <value>, "WidthSamples:=", <value>, "CoordSystem:=", <string name of coordinate system>]</td></tr></table>
Return Value	None.

Python Syntax	InsertRectangleSetup(<*RectangleSetupParams*>)
Python Example	oModule.InsertRectangleSetup(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:Rectangle1", "UseCustomRadiationSurface:=", False, "Length:=" , "20meter", "Width:=" , "20meter", "LengthSamples:=" , 41, "WidthSamples:=" , 41, "CoordSystem:=" , "Global"])

InsertSphereSetup

Inserts a near field sphere radiation setup.

UI Access	Right-click on **Radiation** in the project tree ,select **Insert Near Field Setup** > **Sphere...**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SphereSetupParams*></td><td>Array</td><td>Structured array. ["NAME:<SphereSetupName>", "UseCustomRadiationSurface:=", <boolean>, "Radius:=", "<value><unit>", "ThetaStart:=", "<value><unit>", "ThetaStop:=", "<value><unit>", "ThetaStep:=", "<value><unit>",</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"PhiStart:=", "<value><unit>", "PhiStop:=", "<value><unit>", "PhiStep:=", "<value><unit>", "UseLocalCS:=", <boolean>]
Return Value	None.		

Python Syntax	InsertSphereSetup(<*SphereSetupParams*>)
Python Example	oModule.InsertSphereSetup(["NAME:Sphere2", "UseCustomRadiationSurface:=", False, "Radius:=" , "20meter", "ThetaStart:=" , "0deg", "ThetaStop:=" , "180deg", "ThetaStep:=" , "2deg", "PhiStart:=" , "-180deg", "PhiStop:=" , "180deg", "PhiStep:=" , "2deg", "UseLocalCS:=" , False

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Script Commands for Modifying Antenna Array Setups

EditAntennaArraySetup

EditAntennaArraySetup

Modifies the antenna array setup. There are 3 choices in the setup. The default is set to **No Array Setup**. There are two (other) kinds of arrays that the user can set: **Regular Array Setup** and **Custom Array Setup**.

<table>
<tr><td>UI Access</td><td>Right-click on Radiation in the project tree, select Antenna Array Setup...</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><AntennaArrayParams></td><td>Array</td><td>Structured array.
["NAME:ArraySetupInfo",
"UseOption:=", <string "No Array" | "Parametric Array" | "Custom Array">,
<ParametricArrayParams> | <CustomArrayParams>]</td></tr>
<tr><td><ParametricArrayParams></td><td>Array</td><td>Structured array.
["NAME:ParametricArray",
"DesignFrequency:=", "<value><unit>",
"LayoutType:=", <integer>,
"CenterCellX:=", "<value><unit>",
"CenterCellY:=", "<value><unit>",
"CenterCellZ:=", "<value><unit>",
"SpecifyDesignInWavelength:=", <boolean>,</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"WidthSpacing:=", "<value><unit>",
"WidthSpacingInWavelength:=", "<value>",
"Width:=", "<value><unit>",
"WidthInWavelength:=", "<value>",
"LengthSpacing:=", "<value><unit>",
"LengthSpacingInWavelength:=", "<value>",
"Length:=", "<value><unit>",
"LengthInWavelength:=", "<value>",
"SymmetryType:=", <integer>,
"StaggerAngle:=", "<value><unit>",
"StaggerType:=", <integer>,
"UDirnX:=", "<integer>",
"UDirnY:=", "<integer>",
"UDirnZ:=", "<integer>",
"VDirnX:=", "<integer>",
"VDirnY:=", "<integer>",
"VDirnZ:=", "<integer>",
"WeightType:=", <integer>,
"EdgeTaperX_db:=", "<integer>",
"CosineExp:=", "<integer>",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"DifferentialType:=", <integer>, "Behavior:=", <string>, "ScanAnglePhi:=", "<value><unit>", "ScanAngleTheta:=", "<value><unit>", "UDirnPhaseShift:=", "<value><unit>", "VDirnPhaseShift:=", "<value><unit>"]
	<*CustomArrayParams*>	Array	Structured array. ["NAME:CustomArray", "NumCells:=", <int>, <CellsParamsArray>]
	<*CellsParamsArray*>	Array	Structured array. ["NAME:Cell", <CellParams>, <CellParams>, ...]
	<*CellParams*>	Array	Structured array. ["Name:<CellName>", "XCoord:=", <double>, "YCoord:=", <double>, "ZCoord:=", <double>, "Amplitude:=", <double>, "Phase:=", <double>]
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditAntennaArraySetup(<*AntennaArrayParams*>)
Python Example	oModule.EditAntennaArraySetup(["NAME:ArraySetupInfo", "UseOption:=" , "ParametricArray", ["NAME:ParametricArray", "DesignFrequency:=" , "1GHz", "LayoutType:=" , 1, "CenterCellX:=" , "0mm", "CenterCellY:=" , "0mm", "CenterCellZ:=" , "0mm", "SpecifyDesignInWavelength:=", False, "WidthSpacing:=" , "60mm", "WidthSpacingInWavelength:=", "0.200138457118891", "Width:=" , "480mm", "WidthInWavelength:=" , "1.60110765695113", "LengthSpacing:=" , "60mm", "LengthSpacingInWavelength:=", "0.200138457118891", "Length:=" , "360mm", "LengthInWavelength:=" , "1.20083074271335",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"SymmetryType:="        , 0,
"StaggerAngle:="        , "0deg",
"StaggerType:="         , 0,
"UDirnX:="              , "0",
"UDirnY:="              , "0",
"UDirnZ:="              , "-1",
"VDirnX:="              , "1",
"VDirnY:="              , "0",
"VDirnZ:="              , "0",
"WeightType:="          , 3,
"EdgeTaperX_db:="       , "-200",
"CosineExp:="           , "1",
"DifferentialType:="    , 0,
"Behavior:="            , "UseSlaveSettings",
"ScanAnglePhi:="        , "135deg",
"ScanAngleTheta:="      , "45deg",
"UDirnPhaseShift:="     , "0deg",
"VDirnPhaseShift:="     , "0deg"]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Script Commands for Exporting Antenna Parameters and Max Field Parameters

ExportRadiationParametersToFile

ExportElementPatternToFile

ExportElementPatternToFile

Exports far field data for specified elements to a file.

<table>
<tr><td>UI Access</td><td>Export Element Pattern button in Antenna Parameters dialog box.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><ExportElementPatternFileParams></td><td>Array</td><td>Structured array.

["ExportFileName:=", <string file name include path>,

"SetupName:=" , <string radiation setup to use>,

"IntrinsicVariationKey:=", <string frequency at which to extract the parameters>,

"DesignVariationKey:=", <string design variations at which to extract the parameters.>,

"SolutionName:=", <string solution to use>,

"SourceGroupName:=", <string source group to use>]</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportElementPatternToFile(<*ExportElementPatternFileParams*>)

Python Example	`oModule.ExportElementPatternToFile(`
	`  [`
	`  "ExportFileName:="  , "C:\Documents\\exportelement2.ffd",`
	`  "SetupName:="  , "3D",`
	`  "IntrinsicVariationKey:=", "Freq=\'28GHz\'",`
	`  "DesignVariationKey:="  , "antipad_rad=\'0.14999999999999999mm\'",`
	`  "SolutionName:="  , "Setup1 : LastAdaptive"`
	`  ])`

ExportFieldsToFile

Exports fields to a file.

UI Access	From the **Antenna Parameters** dialog box, **Export Fields** to .ffd file.
Parameters	See table below.

Name	Type	Description

<*ExportParams*>	Array	Structured array.
		`["ExportFileName:=", <string file name include path>,`
		`  "SetupName:=", <string name of the Radiation Far Field or Near Field setup>,`
		`  "IntrinsicVariationKey:=", <string from the Antenna`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Parameters dialog.>, "DesignVariationKey:=", <string>, "SolutionName:=" , <string name of solution to use>, "Quantity:=" , <string antenna Parameter name. If empty, all parameters are exported.>]
Return Value	None.		

Python Syntax	ExportFieldsToFile (<*ExportParams*>)
Python Example	oModule.ExportFieldsToFile(["ExportFileName:=", "C:/exportfields.ffd", "SetupName:=", "Infinite Sphere1", "IntrinsicVariationKey:=", "Freq=\'4.5GHz\'", "DesignVariationKey:=", "", "SolutionName:=", "4500MHz : LastAdaptive", "Quantity:=", ""])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportParametersToFile

Exports antenna parameters to a file.

<table>
<tr><td>UI Access</td><td>Click Export in the Antenna Parameters dialog.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><DefParams></td><td>Array</td><td>Structured array.
<pre>[
  "ExportFileName:=", <string file name include path>,
  "SetupName:=", <string name of setup to export>,
  "IntrinsicVariationKey:=", <string>,
  "DesignVariationKey:=" , <string>,
  "SolutionName:=", <string name of solution to export>,
  "Quantity:=" , <string name of quantity, empty to export all>]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	ExportParametersToFile(<*DefParams*>)

Python Example	`oModule.ExportParametersToFile(`
	`[`
	`  "ExportFileName:=", "C:/exportparams.txt",`
	`  "SetupName:=", "Infinite Sphere1",`
	`  "IntrinsicVariationKey:=", "Freq=\'4.5GHz\'",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"DesignVariationKey:=", "", "SolutionName:=", "4500MHz : LastAdaptive", "Quantity:=", ""])

ExportRadiationFieldsToFile

Exports radiation fields to a file.

UI Access	N/A
Parameters	Name: *<ExportParams>* — Type: Array — Description: Structured array. ["ExportFileName:=", <string file name include path>, "SetupName:=", <string name of the Radiation Far Field or Near Field setup>, "IntrinsicVariationKey:=", <string from the Antenna Parameters dialog.>, "DesignVariationKey:=", <string>, "SolutionName:=" , <string name of solution to use>, "Quantity:=" , <string antenna Parameter name. If empty, all parameters are exported.>]
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportRadiationFieldsToFile (<*ExportParams*>)

Python Example	`oModule.ExportRadiationFieldsToFile(`
	`[`
	`  "ExportFileName:=", "C:/exportfields.ffd",`
	`  "SetupName:=", "Infinite Sphere1",`
	`  "IntrinsicVariationKey:=", "Freq=\'4.5GHz\'",`
	`  "DesignVariationKey:=", "",`
	`  "SolutionName:=", "4500MHz : LastAdaptive",`
	`  "Quantity:=", ""`
	`])`

ExportRadiationParametersToFile

Exports radiation parameters to a file. This command can be used to export the max quantities of a near-field setup and, in the case of far fields, the antenna parameters to the specified file.

UI Access	**HFSS** > **Radiation** > **Compute Max/Antenna Params**

Parameters

Name	Type	Description

<*ExportToFileParams*>	Array	Structured array.
		`["ExportFileName:=", <string file name include path>`
		`  "SetupName:=", <string>`
		`  "IntrinsicVariationKey:=", <string>,`
		`  "DesignVariationKey:=", <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"SolutionName:=", <string>]
Return Value	None.

Python Syntax	ExportRadiationParametersToFile(<*ExportToFileParams*>)
Python Example	oModule.ExportRadiationParametersToFile(["ExportFileName:=", "C:\projects\exportantparams.txt", "SetupName:=", "Infinite Sphere1", "IntrinsicVariationKey:=", "Freq='10GHz'", "DesignVariationKey:=", "", "SolutionName:=", "LastAdaptive"])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

16 - Radiation Setup Manager Module Script Commands

Radiation Setup Manager commands should be executed by the **RadiationSetupMgr** module.

oModule = oDesign.GetModule("RadiationSetupMgr")

oModule.CommandName <args>

The commands in this module create and change radiation boundaries which can be modified by the Radiation module script commands. Radiation boundaries are divided into a far-field boundary in the shape of an infinite sphere and near-field boundaries in the shapes of a sphere, line, or plane. Each boundary uses a different set of parameters. The following commands create and edit the radiation boundaries.

- Add or edit a far field infinite sphere
- Add or edit a near-field line
- Add or edit a near-field plane
- Add or edit a near-field sphere

Add (Infinite Sphere)

Adds a far-field radiation boundary in the shape of an infinite sphere.

> **Note:** Only *Name* and *Type* are required parameters. If you set a custom Theta or Phi parameter, then you must set all Theta and Phi parameters. If you have a local coordinate system, then you must set the *CS* and *UseLocalCS* parameters.

UI Access	In the Project Manager under the design name, right click **Radiation** and in the right-click menu, click **Insert Far Field Setup**>**Infinite Sphere**.
Parameters	

Name	Type	Description
<*Name*>	String	A user-defined name for the infinite sphere.
<*Type*>	String	Type of radiation setup. It should be **IntiniteSphereFarField**.
<*VersionID*>	Integer	Optional. Set to **0**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ID>	Integer	Optional. Identifier that is created differently for every setup.
	<StartTheta >	String	Optional. Starting point for theta in the spherical coordinate system for the sphere being defined. The default value is "**0deg"**. Use the form **"**<num>**deg"**.
	<StartPhi>	String	Optional. Starting point for phi in the spherical coordinate system for the sphere being defined. The default value is **"-180deg"**. Use the form **"**<num>**deg"**.
	<StopTheta>	String	Optional. Stopping point for theta in the spherical coordinate system for the sphere being defined. The default value is **"180deg"**. Use the form **"**<num>**deg"**.
	<StopPhi>	String	Optional. Stopping point for phi in the spherical coordinate system for the sphere being defined. The default value is **"1800deg".** Use the form **"**<num>**deg"**.
	<StepTheta>	String	Optional. Step size between the start and the stop theta values at which the far field is calculated. The default value is **"2deg"**. Use the form **"**<num>**deg"**.
	<StepPhi>	String	Optional. Step size between the start and the stop phi values at which the far field is calculated. The default value is **"2deg"**. Use the form **"**<num>**deg"**.
	<CS>	Integer	Optional. The coordinate system to use. When coordinate system is local, set this number to the index of the local coordinate system. Set this value to **-1** for a global coordinate system. The default value is **-1**.
	<UseLocalCS>	Boolean	Optional. **True** to select a local coordinate system, and **False** to use the global coordinate system. The default value is **False**.
Return Value	None.		

Python Syntax	oModule.Add(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:<*Name*>", "Type:=" , "InfiniteSphereFarField", "VersionID:=" , 0, "ID:=" , <*ID*>, "StartTheta:=" , "<*StartTheta*>deg", "StartPhi:=" , "<*StartPhi*>deg", "StopTheta:=" , "<*StopTheta*>deg", "StopPhi:=" , "<*StopPhi*>deg", "StepTheta:=" , "<*StepTheta*>deg", "StepPhi:=" , "<*StapPhi*>deg", "CS:=" , <*CS*>, "UseLocalCS:=" , <*UseLocalCS*>])

Python Example	`oDesign = oDesign.GetModule("RadiationSetupMgr")`
	`oModule.Add(`
	`[`
	`"NAME:Infinite Sphere 1",`
	`"Type:=", "InfiniteSphereFarField",`
	`"VersionID:=", 0,`
	`"ID:=", 5,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "StartTheta:=", "0deg",
   "StartPhi:=", "-180deg",
   "StopTheta:=", "180deg",
   "StopPhi:=", "180deg",
   "StepTheta:=", "2deg",
   "StepPhi:=", "2deg",
   "CS:=", 1,
   "UseLocalCS:=", False
])
```

Edit (RadiationSetupMgr for Infinite Sphere)

Changes a far-field radiation boundary in the shape of an infinite sphere.

> **Note:** Only *Name* and *Type* are required parameters. If you set a custom Theta or Phi parameter, then you must set all Theta and Phi parameters. If you have a local coordinate system, then you must set the *CS* and *UseLocalCS* parameters.

UI Access	In the Project Manager under the design name > **Radiation**, double click on the name of the infinite sphere.
Parameters	

Name	Type	Description
<*Name*>	String	A user-defined name for the infinite sphere.
<*Type*>	String	Type of radiation setup. It should be **IntiniteSphereFarField**.
<*VersionID*>	Integer	Optional. Set to **0**.
<*ID*>	Integer	Optional. Identifier that is created differently for every setup.
<*StartTheta* >	String	Optional. Starting point for theta in the spherical coordinate system

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			for the sphere being defined. Use the form **"**<num>**deg"**.
	<*StartPhi*>	String	Optional. Starting point for phi in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
	<*StopTheta*>	String	Optional. Stopping point for theta in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
	<*StopPhi*>	String	Optional. Stopping point for phi in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
	<*StepTheta*>	String	Optional. Step size between the start and the stop theta values at which the far field is calculated. Use the form **"**<num>**deg"**.
	<*StepPhi*>	String	Optional. Step size between the start and the stop phi values at which the far field is calculated. Use the form **"**<num>**deg"**.
	<*CS*>	Integer	Optional. The coordinate system to use. When coordinate system is local, set this number to the index of the local coordinate system. Set this value to **-1** for a global coordinate system.
	<*UseLocalCS*>	Boolean	Optional. **True** to select a local coordinate system, and **False** to use the global coordinate system.
Return Value	None.		

Python Syntax

```
oModule.Edit(
    [
        "NAME:<Name>",
        "Type:=" , "InfiniteSphereFarField",
        "VersionID:=" , 0,
        "ID:=" , <ID>,
        "StartTheta:=" , "<StartTheta>deg",
        "StartPhi:=" , "<StartPhi>deg",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"StopTheta:=" , "<*StopTheta*>deg", "StopPhi:=" , "<*StopPhi*>deg", "StepTheta:=" , "<*StepTheta*>deg", "StepPhi:=" , "<*StapPhi*>deg", "CS:=" , <*CS*>, "UseLocalCS:=" , <*UseLocalCS*>])

Python Example	`oDesign = oDesign.GetModule("RadiationSetupMgr")`
	`oModule.Edit(`
	`[`
	`"NAME:Infinite Sphere 1",`
	`"Type:=", "InfiniteSphereFarField",`
	`"VersionID:=", 0,`
	`"ID:=", 5,`
	`"StartTheta:=", "0deg",`
	`"StartPhi:=", "-180deg",`
	`"StopTheta:=", "180deg",`
	`"StopPhi:=", "180deg",`
	`"StepTheta:=", "2deg",`
	`"StepPhi:=", "2deg",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"CS:=", 1, "UseLocalCS:=", False])

Add (Near Field Line)

Adds a near-field radiation boundary in the shape of a line. To calculate the near fields, you must have a 3D line defined across the design area before creating this type of radiation boundary.

UI Access	In the Project Manager under the design name, right click **Radiation** and in the right-click menu, click **Insert Near Field Setup**>**Line**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Name*></td><td>String</td><td>A user-defined name for the near field line.</td></tr><tr><td><*Type*></td><td>String</td><td>Type of radiation setup. It should be **NearFieldLine**.</td></tr><tr><td><*VersionID*></td><td>Integer</td><td>Optional. Set to **0**.</td></tr><tr><td><*ID*></td><td>Integer</td><td>Optional. Identifier that is created differently for every setup.</td></tr><tr><td><*NFLineID* ></td><td>Integer</td><td>Optional. Index of the 3D line in the design area across which the near fields are calculated.</td></tr><tr><td><*NumSamples*></td><td>String</td><td>Optional. Number of points (samples) needed to calculate the near fields across the line. Use the form **"**<num>**"**. The default value is **"1000"**.</td></tr></table>
Return Value	None.

Python Syntax	oModule.Add([

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NAME:<*Name*>", "Type:=" , "NearFieldLine", "VersionID:=" , 0, "ID:=" , <*ID*>, "NFLineID:=", "<*NFLineID*>", "NumSamples:=","<*NumSamples*>"])

Python Example	`oDesign = oDesign.GetModule("RadiationSetupMgr")`
	`oModule.Add(`
	`[`
	`"NAME:Line 1",`
	`"Type:=", "NearFieldLine",`
	`"VersionID:=", 0,`
	`"ID:=", 7,`
	`"NFLineID:=", 312,`
	`"NumSamples:=", "1000"`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit (RadiationSetupMgr for Near Field Line)

Changes a near-field radiation boundary in the shape of a line. To calculate the near fields, you must have a 3D line defined across the design area before creating this type of radiation boundary.

<table>
<tr><td>UI Access</td><td>In the Project Manager under the design name > Radiation, double click on the name of the line.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Name></td><td>String</td><td>A user-defined name for the near field line.</td></tr>
<tr><td><Type></td><td>String</td><td>Type of radiation setup. It should be NearFieldLine.</td></tr>
<tr><td><VersionID></td><td>Integer</td><td>Optional. Set to 0.</td></tr>
<tr><td><ID></td><td>Integer</td><td>Optional. Identifier that is created differently for every setup.</td></tr>
<tr><td><NFLineID ></td><td>Integer</td><td>Optional. Index of the 3D line in the design area across which the near fields are calculated.</td></tr>
<tr><td><NumSamples></td><td>String</td><td>Optional. Number of points (samples) needed to calculate the near fields across the line. Use the form "<num>".</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>
oModule.Edit(
[
"NAME:<<i>Name</i>>",
"Type:=" , "NearFieldLine",
"VersionID:=" , 0,
"ID:=" , <<i>ID</i>>,
"NFLineID:=", "<<i>NFLineID</i>>",
"NumSamples:=","<<i>NumSamples</i>>"
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])
Python Example	oDesign = oDesign.GetModule("RadiationSetupMgr") oModule.Edit(["NAME:Line 1", "Type:=", "NearFieldLine", "VersionID:=", 0, "ID:=", 7, "NFLineID:=", 312, "NumSamples:=", "1000"])

Add (Near Field Plane)

Adds a near-field radiation boundary in the shape of a plane. This plane is calculated in reference to a rectangle defined by the X and Y parameters.

UI Access	In the Project Manager under the design name, right click **Radiation** and in the right-click menu, click **Insert Near Field Setup**>**Plane**.
Parameters	See table below

Name	Type	Description
<*Name*>	String	A user-defined name for the near field sphere.
<*Type*>	String	Type of radiation setup. It should be **NearFieldSphere**.
<*VersionID*>	Integer	Optional. Set to **0**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ID>	Integer	Optional. Identifier that is created differently for every setup.
	<MinX >	String	Optional. Starting X value for the plane that the near fields will be calculated for. The default value is **"-41.5mm"**. Use the form **"**<num><unit>**"**.
	<MaxX>	String	Optional. Final X value for the plane that the near fields will be calculated for. The default value is **"8.5mm"**. Use the form **"**<num><unit>**"**.
	<MinY>	String	Optional. Starting Y value for the plane that the near fields will be calculated for. The default value is **"16mm"**. Use the form **"**<num><unit>**"**.
	<MaxY>	String	Optional. Final Y value for the plane that the near fields will be calculated for. The default value is **"16mm"**. Use the form **"**<num><unit>**"**.
	<Height>	String	Optional. Height of the reference rectangle. The default value is **"1.652145mm"**. Use the form **"**<num>**"**.
	<ExpFactor>	String	Optional. The X and Y dimensions are multipled by this exponential factor to determine new dimensions for the plane. For example, set to **1** to use the default calculated dimensions of the plane or set to 2 to double each dimension of the plane. The default value is **"1"**. Use the form **"**<num>**"**.
	<UseExpFactor>	Boolean	Optional. Set to **True** to add the exponent factor to the plane dimensions or **False** to use the automatically calculated plane dimensions or user-defined plane dimensions. The default value is **False**.
	<CSID>	Integer	Optional. An identifier for the coordinate system.
Return Value	None.		

Python Syntax	oModule.Add(["NAME:<*Name*>",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Type:=" , "NearFieldPlane", "VersionID:=" , 0, "ID:=" , <*ID*>, "MinX:=" , "<*MinX*>", "MaxX:=" , "<*MaxX*>", "MinY:=" , "<*MinY*>", "MaxY:=" , "<*MaxY*>", "Height:=" , "<*Height*>", "ExpFactor:=" , "<*ExpFactor*>", "UseExpFactor:=" , <*UseExpFactor*>, "CSID:=" , <*CSID*>])

Python Example	`oDesign = oDesign.GetModule("RadiationSetupMgr")`
	`oModule.Add(`
	`[`
	`"NAME:Plane 1",`
	`"Type:=", "NearFieldPlane",`
	`"VersionID:=", 0,`
	`"ID:=", 7,`
	`"MinX:=", "-50mm",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "MaxX:=", "25mm",
   "MinY:=", "-18.5mm",
   "MaxY:=", "31.5mm",
   "Height:=", "5.6mm",
   "ExpFactor:=", "1",
   "UseExpFactor:=", False,
   "CSID:=", -4
])
```

Edit (RadiationSetupMgr for Near Field Plane)

Changes a near-field radiation boundary in the shape of a plane. This plane is calculated in reference to a rectangle defined by the X and Y parameters.

UI Access

In the Project Manager under the design name > **Radiation**, double click on the name of the plane.

Parameters

Name	Type	Description
<*Name*>	String	A user-defined name for the near field sphere.
<*Type*>	String	Type of radiation setup. It should be **NearFieldSphere**.
<*VersionID*>	Integer	Optional. Set to **0**.
<*ID*>	Integer	Optional. Identifier that is created differently for every setup.
<*MinX* >	String	Optional. Starting X value for the plane that the near fields will be calculated for. Use the form **"**<num><unit>**"**.
<*MaxX*>	String	Optional. Final X value for the plane that the near fields will be calculated for. Use the form **"**<num><unit>**"**.
<*MinY*>	String	Optional. Starting Y value for the plane that the near fields will be calculated for. Use the form **"**<num><unit>**"**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*MaxY*>	String	Optional. Final Y value for the plane that the near fields will be calculated for. Use the form **"**<num><unit>**"**.
	<*Height*>	String	Optional. Height of the reference rectangle. Use the form **"**<num>**"**.
	<*ExpFactor*>	String	Optional. The X and Y dimensions are multipled by this exponential factor to determine new dimensions for the plane. For example, set to **1** to use the default calculated dimensions of the plane or set to 2 to double each dimension of the plane. Use the form **"**<num>**"**.
	<*UseExpFactor*>	Boolean	Optional. Set to **True** to add the exponent factor to the plane dimensions or **False** to use the automatically calculated plane dimensions or user-defined plane dimensions.
	<*CSID*>	Integer	Optional. An identifier for the coordinate system.
Return Value	None.		

Python Syntax

```
oModule.Edit(
    [
        "NAME:<Name>",
        "Type:=" , "NearFieldPlane",
        "VersionID:=" , 0,
        "ID:=" , <ID>,
        "MinX:=" , "<MinX>",
        "MaxX:=" , "<MaxX>",
        "MinY:=" , "<MinY>",
        "MaxY:=" , "<MaxY>",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Height:=" , "<*Height*>", "ExpFactor:=" , "<*ExpFactor*>", "UseExpFactor:=" , <*UseExpFactor*>, "CSID:=" , <*CSID*>])
Python Example	oDesign = oDesign.GetModule("RadiationSetupMgr") oModule.Edit(["NAME:Plane 1", "Type:=", "NearFieldPlane", "VersionID:=", 0, "ID:=", 7, "MinX:=", "-50mm", "MaxX:=", "25mm", "MinY:=", "-18.5mm", "MaxY:=", "31.5mm", "Height:=", "5.6mm", "ExpFactor:=", "1", "UseExpFactor:=", False, "CSID:=", -4])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Add (Near Field Sphere)

Adds a near-field radiation boundary in the shape of a sphere.

> **Note:** Only *Name* and *Type* are required parameters. If you set a custom Theta or Phi parameter, then you must set all Theta and Phi parameters. If you have a local coordinate system, then you must set the *CS* and *UseLocalCS* parameters.

<table>
<tr><td>UI Access</td><td>In the Project Manager under the design name, right click Radiation and in the right-click menu, click Insert Near Field Setup>Sphere.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><<i>Name</i>></td><td>String</td><td>A user-defined name for the near field sphere.</td></tr>
<tr><td><<i>Type</i>></td><td>String</td><td>Type of radiation setup. It should be NearFieldSphere.</td></tr>
<tr><td><<i>VersionID</i>></td><td>Integer</td><td>Optional. Set to 0.</td></tr>
<tr><td><<i>ID</i>></td><td>Integer</td><td>Optional. Identifier that is created differently for every setup.</td></tr>
<tr><td><<i>StartTheta</i> ></td><td>String</td><td>Optional. Starting point for theta in the spherical coordinate system for the sphere being defined. The default value is "0deg". Use the form "<num>deg".</td></tr>
<tr><td><<i>StartPhi</i>></td><td>String</td><td>Optional. Starting point for phi in the spherical coordinate system for the sphere being defined. The default value is "-180deg". Use the form "<num>deg".</td></tr>
<tr><td><<i>StopTheta</i>></td><td>String</td><td>Optional. Stopping point for theta in the spherical coordinate system for the sphere being defined. The default value is "180deg". Use the form "<num>deg".</td></tr>
<tr><td><<i>StopPhi</i>></td><td>String</td><td>Optional. Stopping point for phi in the spherical coordinate system for the sphere being defined. The default value is "1800deg". Use the form "<num>deg".</td></tr>
<tr><td><<i>StepTheta</i>></td><td>String</td><td>Optional. Step size between the start and the stop theta values at</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			which the far field is calculated. The default value is **"2deg"**. Use the form **"**<num>**deg"**.
	<*StepPhi*>	String	Optional. Step size between the start and the stop phi values at which the far field is calculated. The default value is **"2deg"**. Use the form **"**<num>**deg"**.
	<*CS*>	Integer	Optional. The coordinate system to use. When coordinate system is local, set this number to the index of the local coordinate system. Set this value to **-1** for a global coordinate system. The default value is **-1**.
	<*UseLocalCS*>	Boolean	Optional. **True** to select a local coordinate system, and **False** to use the global coordinate system. The default value is **False**.
	<*NFRadius*>	String	Optional. Radius of the sphere we want to calculate near fields on. The default value is **"20mm"**.Use the form **"**<num><unit>**"**.
Return Value	None.		

Python Syntax

```
oModule.Add(
    [
        "NAME:<Name>",
        "Type:=" , "InfiniteSphereFarField",
        "VersionID:=" , 0,
        "ID:=" , <ID>,
        "StartTheta:=" , "<StartTheta>deg",
        "StartPhi:=" , "<StartPhi>deg",
        "StopTheta:=" , "<StopTheta>deg",
        "StopPhi:=" , "<StopPhi>deg",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"StepTheta:=" , "<*StepTheta*>deg", "StepPhi:=" , "<*StapPhi*>deg", "CS:=" , <*CS*>, "UseLocalCS:=" , <*UseLocalCS*>, "NFRadius:=", "<*NFRadius*>"])

Python Example	`oDesign = oDesign.GetModule("RadiationSetupMgr")`
	`oModule.Add(`
	`[`
	`"NAME:Sphere 1",`
	`"Type:=", "NearFieldSphere",`
	`"VersionID:=", 0,`
	`"ID:=", 7,`
	`"StartTheta:=", "0deg",`
	`"StartPhi:=", "-180deg",`
	`"StopTheta:=", "180deg",`
	`"StopPhi:=", "180deg",`
	`"StepTheta:=", "2deg",`
	`"StepPhi:=", "2deg",`
	`"CS:=", -1,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "UseLocalCS:=", False,
   "NFRadius:=", "20mm"
])
```

Edit (RadiationSetupMgr for Near Field Sphere)

Changes a near-field radiation boundary in the shape of a sphere.

Note: Only *Name* and *Type* are required parameters. If you set a custom Theta or Phi parameter, then you must set all Theta and Phi parameters. If you have a local coordinate system, then you must set the *CS* and *UseLocalCS* parameters.

UI Access

In the Project Manager under the design name > **Radiation**, double click on the name of the near-field sphere.

Parameters

Name	Type	Description
<*Name*>	String	A user-defined name for the near field sphere.
<*Type*>	String	Type of radiation setup. It should be **NearFieldSphere**.
<*VersionID*>	Integer	Optional. Set to **0**.
<*ID*>	Integer	Optional. Identifier that is created differently for every setup.
<*StartTheta* >	String	Optional. Starting point for theta in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
<*StartPhi*>	String	Optional. Starting point for phi in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
<*StopTheta*>	String	Optional. Stopping point for theta in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
<*StopPhi*>	String	Optional. Stopping point for phi in the spherical coordinate system for the sphere being defined. Use the form **"**<num>**deg"**.
<*StepTheta*>	String	Optional. Step size between the start and the stop theta values at which the far field is calculated. Use the form **"**<num>**deg"**.
<*StepPhi*>	String	Optional. Step size between the start and the stop phi values at

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			which the far field is calculated. Use the form **"**<num>**deg"**.
	<*CS*>	Integer	Optional. The coordinate system to use. When coordinate system is local, set this number to the index of the local coordinate system. Set this value to **-1** for a global coordinate system.
	<*UseLocalCS*>	Boolean	Optional. **True** to select a local coordinate system, and **False** to use the global coordinate system.
	<*NFRadius*>	String	Optional. Radius of the sphere we want to calculate near fields on. Use the form **"**<num><unit>**"**.
Return Value	None.		

Python Syntax	oModule.Edit(["NAME:<*Name*>", "Type:=" , "InfiniteSphereFarField", "VersionID:=" , 0, "ID:=" , <*ID*>, "StartTheta:=" , "<*StartTheta*>deg", "StartPhi:=" , "<*StartPhi*>deg", "StopTheta:=" , "<*StopTheta*>deg", "StopPhi:=" , "<*StopPhi*>deg", "StepTheta:=" , "<*StepTheta*>deg", "StepPhi:=" , "<*StapPhi*>deg",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"CS:=" , <*CS*>, "UseLocalCS:=" , <*UseLocalCS*>, "NFRadius:=", "<*NFRadius*>"])
Python Example	oDesign = oDesign.GetModule("RadiationSetupMgr") oModule.Edit(["NAME:Sphere 1", "Type:=", "NearFieldSphere", "VersionID:=", 0, "ID:=", 7, "StartTheta:=", "0deg", "StartPhi:=", "-180deg", "StopTheta:=", "180deg", "StopPhi:=", "180deg", "StepTheta:=", "2deg", "StepPhi:=", "2deg", "CS:=", -1, "UseLocalCS:=", False, "NFRadius:=", "20mm"])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

17 - User-Defined Document Script Commands

The product has to implement the GetModule call to create the UserDefinedDocument scripting object (e.g., Check AltraSimDesign.cpp (function GetMgrIDispatch()). To access the UserDefinedDocuments scripting object, use:

```
oModule = oDesign.GetModule("UserDefinedDocuments")
```

Once you have the scripting object, you can use the following methods:

- AddDocument
- EditDocument
- RenameDocument
- DeleteDocument
- UpdateDocument
- ViewHtmlDocument
- ViewPdfDocument
- SaveHtmlDocumentAs
- SavePdfDocumentAs
- GetDocumentDefinitionNames
- DeleteAllDocuments
- UpdateAllDocuments

You can find examples of how to use these methods on each of the methods' pages. A complete example of a Python script is also available, as is an example with a line by line explanation.

AddDocument

Creates a new document based on provided data and traces. The document names come from UserDefinedDocument folder under syslib, userlib, and personallib. This creates a document and places it in the Project Manager under **Results** > **Documents**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right click on **Results** > **Create Document**. Choose a document name.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><data></td><td>Array</td><td>Data the defines the document.</td></tr><tr><td><traces></td><td>Array</td><td>trace data for the inputs in the document.</td></tr></table>
Return Value	None

Python Syntax	AddDocument (<*data*>, <*traces*>)

Python Example	`oModule.AddDocument(`
	`  ["NAME:Design Summary",`
	`  "",`
	`  "SysLib",`
	`  "DesignSummary",`
	`  ["NAME:Inputs"]`
	`  ],`
	`  ["NAME:DocTraces"]`
	`)`

DeleteAllDocuments

Deletes all documents in the object.

UI Access	Right click on **Documents** in the Project Manager and click **Delete All Documents**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	None.

Python Syntax	DeleteAllDocuments()
Python Example	`oModule.DeleteAllDocuments()`

DeleteDocument

Deletes a specified document.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Delete**.
Parameters	Name | Type | Description <name> | String | Name of the document to be deleted.
Return Value	None.

Python Syntax	DeleteDocument(<*names*>)
Python Example	`oModule.DeleteDocument("Project Design Summary")`

EditDocument

Edits specified documents. If the document pops up a dialog box, the user can make a change the inputs for the document. The document is regenerated and updated. A new one is *not* created.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Modify document**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><originalName></td><td>String</td><td>Name of the original document</td></tr><tr><td><modifiedData></td><td>Array</td><td>New data to add to or modify the document</td></tr><tr><td><modifiedraces></td><td>Array</td><td>Trace data for the inputs of the document</td></tr></table>
Return Value	None.

Python Syntax	EditDocument(<*name*>,<*data*>,<*traces*>)

Python Example	`oModule.EditDocument("Design Summary",`
	`[`
	`  "NAME:Design Summary",`
	`  "",`
	`  "SysLib",`
	`  "DesignSummary",`
	`  ["NAME:Inputs"]`
	`],`
	`["NAME:DocTraces"]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDocumentDefinitionNames

Document definition names are the list of names that can be used to create a document. They appear when you click on **Create document**. This method returns the filenames of document definitions according to the files in the installation directories:

- syslib/UserDefinedDocuments
- userlib/UserDefinedDocuments
- personalllb/UserDefinedDocuments

UI Access	NA
Parameters	Name | Type | Description <separator> | String | Separator used to convey the directory "level"
Return Value	None.

Python Syntax	GetDocumentDefinitionNames(<*separator*>)
Python Example	`oModule.GetDocumentDefinitionNames("/")`

GetDocumentNames

Retrieves the names for all documents.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing document names.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetDocumentNames()
Python Example	`oModule.GetDocumentNames()`

RenameDocument

Changes the name of a document.

UI Access	Right click on the created document in the Project Manager under **Results**> **Documents** and click **Rename**.
Parameters	Name | Type | Description <oldName> | String | Current name of the document <newName> | String | New name of the document
Return Value	None.

Python Syntax	RenameDocument(<*oldName*>, <*newName*>)
Python Example	`oModule.RenameDocument("Design Summary", "Project Design Summary")`

SaveHtmlDocumentAs

Saves a pre-existing HTML file to a different name and/or location.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Save As** > **HTML**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name | Type | Description *<name>* | String | Name of the document to be saved. *<saveTo>* | String | File path to save the document to.
Return Value	none

Python Syntax	SaveHtmlDocumentAs(*<name>*, *<saveTo>*)
Python Example	`oModule.SaveHtmlDocumentAs("Design Summary 1", "DS1.html")`

SavePdfDocumentAs

Saves a pre-existing PDF file to a different name and/or location.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Save As** > **PDF**.
Parameters	Name | Type | Description *<name>* | String | Name of the document to be saved. *<saveTo>* | String | File path to save the document at.
Return Value	None.

Python Syntax	SavePdfDocumentAs(*<name>*, *<saveTo>*)
Python Example	`oModule.SavePdfDocumentAs("Design Summary 1", "DS1.pdf")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateAllDocuments

Refreshes the contents of all created documents. This action is made on the folder rather than the individual document.

UI Access	Right click on **Results** > **Documents** in the Project Manager and click **Update All Documents**.
Parameters	None.
Return Value	None.

Python Syntax	UpdateAllDocuments()
Python Example	`oModule.UpdateAllDocuments()`

UpdateDocument

Refreshes the contents of the selected document.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Update Document**.
Parameters	Name / Type / Description *<name>* / String / Name of the document to be updated
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	UpdateDocument(<*name*>)
Python Example	`oModule.UpdateDocument("Test UDD Report")`

ViewHtmlDocument

Displays a pre-existing document as HTML.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **View Xml Document**.
Parameters	Name | Type | Description <*name*> | String | Name of the document to be viewed as a HTML
Return Value	None

Python Syntax	ViewHtmlDocument(<*name*>)
Python Example	`oModule.ViewHtmlDocument("Design Summary 1")`

ViewPdfDocument

Displays a pre-existing document as a PDF file.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **View PDF Document**.
Parameters	Name | Type | Description <*name*> | String | Name of the document to be viewed as a PDF

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ViewPdfDocument(<*name*>)
Python Example	`oModule.ViewPdfDocument("Design Summary 1")`

Example Python Script: Defining a Document

This script creates a user-defined solution and a document.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("BJT_Inverter")
oDesign = oProject.SetActiveDesign("Nexxim1")
oModule = oDesign.GetModule("UserDefinedSolutionModule")

oModule.CreateUserDefinedSolution("UDS Distance Trace Arithmetic Result1", "SysLib", "TraceArith-
metic/Distance Sweep Trace Arithmetic",
       [
              "Offset 1:=", "0",
              "Scale 1:=", "1",
              "Offset 2:=", "0",
              "Scale 2:=", "1",
              "Operation:=", "Add"
       ],
       [
              [
                     "Standard",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "probe1",
                        "Transient",
                        [
                                style="font-family: monospace;">"NAME:Context",
                                style="font-family: monospace;">"SimValueContext:=", [1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"500000000","DT",False,"0.001","NUMLEVELS",False,"0",-
"WE",False,"10us","WM",False,"10us","WN",False,"0ns","WS",False,"0ns"]
                        ],
                        [
                                "Time:=", ["All"]
                        ],
                        [
                                "Probe Component:=", ["V(Port1)"]
                        ],
                        []
                ],
                [
                        "Standard",
                        "probe2",
                        "Transient",
                        [
                                "NAME:Context",
                                "SimValueContext:=", [1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"500000000","DT",False,"0.001","NUMLEVELS",False,"0",-
"WE",False,"10us","WM",False,"10us","WN",False,"0ns","WS",False,"0ns"]
                        ],
                        [
                                "Time:=", ["All"]
                        ],
                        [
                                "Probe Component:=", ["V(Port1)"]
                        ],
                        []
                ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [])
oModule = oDesign.GetModule("UserDefinedDocuments")
oModule.AddDocument(
        [
                "NAME:Test Report",
                "Test Report",
                "SysLib",
                "TestUDDInputs",
                [
                        "NAME:Inputs",
                        [
                                "NAME:UDS1",
                                "Solution",
                                "UDS Distance Trace Arithmetic Result1",
                                1000000,
                                0
                        ],
                        [
                                "NAME:UDS2",
                                "Solution",
                                "UDS Distance Trace Arithmetic Result2",
                                1000000,
                                2
                        ]
                ]
        ],
        [
                "NAME:DocTraces",
                [
                        "NAME:UDS1",
                        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                "User Defined",
                                "",
                                "UDS Distance Trace Arithmetic Result1",
                                [
                                        "Context:=", ""
                                ],
                                [
                                        "Distance:=", ["All"]
                                ],
                                [
                                        "Probe Component:=", [""]
                                ],
                                []
                        ]
                ],
                [
                        "NAME:UDS2",
                        [
                                "User Defined",
                                "",
                                "UDS Distance Trace Arithmetic Result1",
                                [
                                        "Context:=", ""
                                ],
                                [
                                        "Distance:=", ["All"]
                                ],
                                [
                                        "Probe Component:=", [""]
                                ],
                        []
                ]
        ]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

18 - User-Defined Solutions Commands

User-Defined Solution commands should be executed by the "UserDefinedSolutionModule" module.

```
oDesign = oProject.SetActiveDesign("TestDesign1")
oModule = oDesign.GetModule("UserDefinedSolutionModule")
```

The list of commands is as follows:

CreateUserDefinedSolution

DeleteUserDefinedSolutions

EditUserDefinedSolution

GetUserDefinedSolutionNames

GetUserDefinedSolutionProperties

CreateUserDefinedSolution

Creates a new user defined solution.

UI Access	Right-click on **Results** > **Create User Defined Solution**.		
Parameters	Name	Type	Description
	<*SoluName*>	String	Name of user defined solution.
	<*LibType*>	String	Indicates the library where the UDS plugin file is located. This parameter must be one of the following values: "SysLib", "UserLib", "PersonalLib".
	<*RelativePath*>	String	The path of the UDS plugin file relative to the "UserDefinedOutputs" sub-directory of the library specified by <*LibType*>.
	<*PropList*>	Array	Strings specify name-value pairs corresponding to the UDS properties specified in the plugin file. For example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			`["multiply_factor:=", "2.0", "component_name:=", "resistor1"]`
	<*ProbeSelections*>	Array	Name of the probe being specified. Note: this must match a probe name specified in the UDS plugin file.
	<DynamicProbes>	Array	Array of <ProbeSelection>'s, representing the probes that are used by dynamic-probes.
Return Value	String name of created user defined solution.		

Python Syntax	CreateUserDefinedSolution(<*SoluName*>, <*LibType*>, <*RelativePath*>, <*PropList*>, <*ProbeSelections*>, <*DynamicProbes*>)

Python Example	`oModule.CreateUserDefinedSolution(`
	`"ConstantTimestep1", "SysLib",`
	`"ConstantTimestep", [], [], [])`

DeleteUserDefinedSolutions

Deletes one or more user defined solutions.

UI Access	**Delete** button from the **User Defined Solutions** dialog.		
Parameters	Name	Type	Description
	<*SoluNames*>	Array	Array of strings containing names of User Defined Solutions to be deleted.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteUserDefinedSolutions(<*SoluNames*>)
Python Example	`oModule.DeleteUserDefinedSolutions(["Solution1", "Solution2"])`

EditUserDefinedSolution

Modifies an existing user defined solution.

UI Access	**Edit** button from the **User Defined Solutions** dialog box.
Parameters	

Name	Type	Description
<*SoluName*>	String	Name of user defined solution to be edited.
<*NewSoluName*>	String	New name for the specified user defined solution.
<*LibType*>	String	Indicates the library where the UDS plugin file is located. This parameter must be one of the following values: "SysLib", "UserLib", "PersonalLib".
<*RelativePath*>	String	The path of the UDS plugin file relative to the "UserDefinedOutputs" sub-directory of the library specified by <*LibType*>.

<*PropList*>	Array	Strings specify name-value pairs corresponding to the UDS properties specified in the plugin file.
		`For example:`
		`["multiply_factor:=", "2.0", "component_name:=", "resistor1"]`

<*ProbeSelections*>	Array	Name of the probe being specified. Note: this must match a probe name specified in the UDS plugin file.
<DynamicProbes>	Array	Array of <ProbeSelection>'s, representing the probes that are used by dynamic-probes.

Return Value	String name of update user defined solution.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditUserDefinedSolution(<*SoluName*>, <*NewSoluName*>, <*LibType*>, <*RelativePath*>, <*PropList*>, <*ProbeSelections*>, <*DynamicProbes*>)

Python Example	`oModule.EditUserDefinedSolution("ConstantTimestep1"`
	`"ConstantTimestep1After", "SysLib",`
	`"ConstantTimestep", [], [], [])`

GetUserDefinedSolutionNames

Retrieves user defined solution names.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing solution names.

Python Syntax	GetUserDefinedSolutionNames()
Python Example	`oModule.GetUserDefinedSolutionNames()`

GetUserDefinedSolutionProperties

Obtains properties for a specified user defined solution.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluName*></td><td>String</td><td>Name of a specified user defined solution.</td></tr></table>
Return Value	Array of strings containing properties values.

Python Syntax	GetUserDefinedSolutionProperties(<*SoluName*>)
Python Example	`oModule.GetUserDefinedSolutionProperties("ConstantTimestep1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

19 - Network Data Explorer Script Commands

Network Data Explorer (NDE) scripting uses three objects, each with unique script commands:

- **Network Data Explorer** – top-level object obtained by calling oNDE=oDesktop.GetTool("ndExplorer"). Network Data Explorer commands are called using `oNDE`.
- **Network Data** – single set of S-parameters, corresponding to a single entry in the UI tree. Network Data commands are called using `oData`.
- **Post Process Settings** – settings that can be applied and removed from network data without making permanent changes to the underlying data. Post Process Settings commands are called using `oPostProc`.

Examples using the above objects:

```
oNDE = oDesktop.GetTool("ndExplorer")

oData = oNDE.Open("D:\folder\test.s2p")

success = oPostProc.AddDiffPair(2, 1, "Diff1", "Comm1", 100, 25)
```

The following commands are described in this section:

AddDiffPair

Cascade (SPISim)

ClearDiffPairs

Clone

Close

Combine (SPISim)

Deembed (SPISim)

DeembedBack (SPISim)

DeembedFront (SPISim)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DisableDiffPairs

EnableDiffPairs

ExportCitiFile

ExportFullWaveSpice

ExportMatlab

ExportNMFData

ExportNetworkData

ExportSpreadsheet

ExportTouchstone

ExportTouchstone2

Extract (SPISim)

GetFrequencies

GetFrequencyCount

GetName

GetPortCount

GetPortNumber

GetPostProcSettings

GetSolutionVariation

GetVariation

HasSameData

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

LoadSolution

Open

Rename (SPISim)

Renormalize (SPISim)

Reorder

Reorder (SPISim)

Reset

SetAllPortImpedances

SetAllPortImpedances

SetPortDeembedDistance

SetPortImpedance

SetPostProcSettings

Smooth

Stretch (SPISim)

Terminate

> **Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.**

AddDiffPair

Specifies a differential pair from two terminal ports.

UI Access	From the **Network Data Explorer** tab, select **Differential Pairs** in the **NDE** ribbon to open the **Differential**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td></td><td>Pairs window. Then select differential pairs from the Pairs group box and click Add Pairs >>.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><positiveTerminal></td><td>Integer</td><td>The portNumber of the positive terminal.</td></tr>
<tr><td><negativeTerminal></td><td>Integer</td><td>The portNumber of the negative terminal.</td></tr>
<tr><td><diffName></td><td>String</td><td>The new name of the differential pin (e.g., Diff1).</td></tr>
<tr><td><commName></td><td>String</td><td>The new name of the common pin (e.g., Comm1).</td></tr>
<tr><td><diffImpedance></td><td>Double</td><td>The differential pin characteristic impedance.</td></tr>
<tr><td><commImpedance></td><td>Double</td><td>The common pin characteristic impedance.</td></tr>
<tr><td><matched></td><td>Boolean</td><td>Specify whether to use matched pair.
Note: The default value is False.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Boolean.</td></tr>
</table>

Python Syntax	AddDiffPair()
Python Example	`success = oPostProc.AddDiffPair(2, 1, "Diff1", "Comm1", 100, 25)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Cascade (SPISim)

Creates new network data from a group of network data objects cascaded together. The network data must have an even number of terminal ports and must have the same number of ports.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

> **Note:** Cascade opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Cascade** from the drop-down menu.
Parameters	**Name** / **Type** / **Description** *<dataArray>* / Array / These network data objects are cascaded together to create the new network data.
Return Value	Network IDispatch. > **Note:** Cascade returns **None** if the specified network data does not have compatible terminal ports defined.

Python Syntax	Cascade()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Cascade([oData1, oData2, oData3])`

> **Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.**

ClearDiffPairs

Clears all differential pair definitions. All other post-processing settings remain the same.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	From the **Network Data Explorer** tab, select **Differential Pairs** in the **NDE** ribbon to open the **Differential Pairs** window. Then select differential pairs from the rightmost box and click **<< Remove Pairs**.
Parameters	None.
Return Value	None.

Python Syntax	ClearDiffPairs()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.ClearDiffPairs()`

> **Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.**

Clone

Creates a copy of the network data object.

UI Access	From the **Network Data Explorer** tab, select the network data object to clone. Then select **Clone** from the **NDE** ribbon.
Parameters	None.
Return Value	Network IDispatch.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Clone()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData1 = oData.Clone()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Close

Closes the network data object. The object will no longer be accessible.

UI Access	From the **Network Data Explorer** tab, click **Close** in the **NDE** ribbon, or select **File** > **Close**.
Parameters	None.
Return Value	None.

Python Syntax	Close()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData.Close()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Combine (SPISim)

Combines frequencies from multiple network data objects to create a new network data with all of the frequencies.

> **Note:** Combine opens and interacts with **SPISim** to complete.

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select Transforms in the NDE ribbon. Then select Combine from the drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><dataArray></td><td>Array</td><td>These network data objects are combined to create the new network data.</td></tr>
<tr><td><freqTol></td><td>Double</td><td>If abs(f1 – f2) < freqTol, f1 and f2 are considered the same frequency.
Note: The default value is 100.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Network IDispatch.
Note: Combine returns None if the network data does not have the same number of ports.</td></tr>
</table>

Python Syntax	Combine()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData4 = oNDE.Combine([oData1, oData2, oData3], 1000)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Deembed (SPISim)

Creates a new network data, dembedding the frontData and the backData from the front and back of the originalData.

Note: Deembed opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Deembed** from the drop-down menu to open the **Deembed...** window, and choose **Given 3 S-params in order of: Total, Front, Back** from the **Settings** group box.
Parameters	**Name** | **Type** | **Description** *<originalData>* | Object | Original network data that is deembeded. *<frontData>* | Object | Network data that is deembeded from the front of the original. *<backData>* | Object | Network data that is deembeded from the back of the original.
Return Value	Network IDispatch. **Note:** Deembed returns **None** if the specified network data does not have compatible terminal ports defined.

Python Syntax	Deembed()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oData2 = oNDE.Deembed(oData1, oDataFront, oDataBack)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

DeembedBack (SPISim)

Creates a new network data, dembedding the backData from the back of the originalData.

Note: DeembedBack opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Deembed** from the drop-down menu to open the **Deembed...** window, and choose **Given 3 S-params in order of: Total, Back** from the **Settings** group box.
Parameters	**Name** | **Type** | **Description** *<originalData>* | Object | Original network data that is deembeded. *<backData>* | Object | Network data that is deembeded from the back of the original.
Return Value	Network IDispatch. **Note:** DeembedBack returns **None** if the specified network data does not have compatible terminal ports defined.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeembedBack()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.DeembedBack(oData1, oDataBack)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

DeembedFront (SPISim)

Creates a new network data, dembedding the frontData from the front of the originalData.

Note: DeembedFront opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Deembed** from the drop-down menu to open the **Deembed...** window, and choose **Given 3 S-params in order of: Total, Front** from the **Settings** group box.
Parameters	**Name** / **Type** / **Description** *<originalData>* / Object / Original network data that is deembeded. *<frontData>* / Object / Network Data that is deembeded from the front of the original.
Return Value	Network IDispatch. **Note:** DeembedFront returns **None** if the specified network data does not have compatible terminal ports defined.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeembedFront()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.DeembedFront(oData1, oDataFront)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

DisableDiffPairs

Deactivates all differential pair definitions. This script does not remove them, so they can be reactivated.

UI Access	From the **Network Data Explorer** tab, select **Differential Pairs** in the **NDE** ribbon to open the **Differential Pairs** window. Once differential pairs have been added to the rightmost box (i.e., select differential pairs from the Pairs group box and click **Add Pairs >>**), remove check marks from the **Enabled** column to deactivate the corresponding differential pairs.
Parameters	None.
Return Value	None.

Python Syntax	DisableDiffPairs()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.DisableDiffPairs()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

EnableDiffPairs

Enables all differential pair definitions.

UI Access	From the **Network Data Explorer** tab, select **Differential Pairs** in the **NDE** ribbon to open the **Differential Pairs** window. Once differential pairs have been added to the rightmost box (i.e., select differential pairs from the Pairs group box and click **Add Pairs >>**), add check marks to the **Enabled** column to activate the corresponding differential pairs.
Parameters	None.
Return Value	None.

Python Syntax	EnableDiffPairs()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oPostProc.EnableDiffPairs()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

ExportCitiFile

Writes the S-parameters to disk in Citifile format (*.cit text file).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select File > Save As to open a Save As explorer window. Then select Citifile (*.cit) from the Save as type: drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><filePath></td><td>String</td><td>The full path to the file.</td></tr>
<tr><td><matrixType></td><td>Integer</td><td>One of the following:
0 – S-Parameters
1 – Y-Parameters
2 – Z-Parameters
3 – Gamma
4 – Impedance
Note: The default value is 0.</td></tr>
<tr><td><formalType></td><td>Integer</td><td>One of the following:
0 – Magnitude/Angle (degrees)
1 – Real/Imaginary
2 – dB/Angle (degrees)
Note: The default value is 0.</td></tr>
<tr><td><precision></td><td>Integer</td><td>The number of significant figures.
Note: The default value is 10.</td></tr>
<tr><td><frequencies></td><td>Array of Doubles</td><td>A subset of the calculated frequencies. Any array values that don’t have calculated data will be ignored. An empty array will cause all frequencies to be written.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: The default value is an empty array.
Return Value	Boolean True if file was successfully written; else False.		

Python Syntax	ExportCitiFile()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.ExportCitiFile("D:\folder\ne133.cit", 0, 1, 12, [])`

ExportFullWaveSpice

Export FullWaveSpice data in a format of your choice.

UI Access	**File** > **Export MacroModel** > **Broadband** (SYZ, FWS….)		
Parameters	Name	Type	Description
	DesignName	String	Design name. Can be left blank if loading solution from a file.
	true/false	Bool	true - solution loaded from file; false - loaded from design
	Name	String	If loading from design, this is the solution name; else, full path of the file
	variation		Pick a particular variation. Leave blank if none.
	["NAME:Frequencies"]	Array	Optional; if none defined, all frequencies are used
	["NAME:SpiceData"]	Array	Spice export options object. The following parameters are for this array. SpiceType: "SSS" - Options: "PSpice", "HSpice", "Spectre", "SSS", "Simplorer", "TouchStone1.0", "TouchStone2.0"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			EnforcePassivity	false - Enforce passivity (true/false)
			EnforceCausality	false - Enforce causality (true/false)
			UseCommonGround	false - Use common ground (true/false)
			FittingError	0.5 - Fitting error
			MaxPoles	400 - Maximum order
			PassivityType	"ConvexOptimization" - Options: "ConvexOptimization", "PassivityByPerturbation", "IteratedFittingOfPV", "IteratedFittingOfPVLF"
			ColumnFittingType	"Column" - Options: "Column", "Entry", "Matrix"
			SSFittingType	"TWA" - Options: "TWA", "IterativeRational"
			RelativeErrorToleranc	false - Relative error tolerance (true/false)
			TouchstoneFormat	"MA" - Options: "MA", "RI", "DB"
			TouchstoneUnits	"Hz" - Options: "Hz", "KHz", "MHz"
			TouchStonePrecision	8 - Touchstone precision
			ExportDirectory	"C:/Examples/LNA/" - Directory to export to
			ExportSpiceFileName	"Linckt_HBTest_2.sss" - Spice export file name
			FullwaveSpiceFileName	"Linckt_HBTest.sss" - Fullwave Spice file name
			CreateNPortModel	true - Create a model based on the exported file (true/false)
Return Value	[none]			

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportFullWaveSpice(<DesignName> , <True/False> , <Name> , <Variation> , <[NAME:Frequencies]> , <[NAME:SpiceData]>)
Python Example	oTool.ExportFullWaveSpice("", False, "", "$isolation=\'0.13\' p1=\'8.7062500000000007mm\' p2=\'8.6100700000000003mm\' s1=\'0.41463699999999998mm\' s2=\'1.40411mm\' w1=\'0.44363399999999997mm\' w2=\'0.47960799999999998mm\' w50=\'1.0031099999999999mm\'", ["NAME:Frequencies", 4500000000, 4505000000, 4510000000, 4515000000, 4520000000, 4525000000, 4530000000, 4535000000, 4540000000, 4545000000], ["NAME:SpiceData", "SpiceType:=" , "HSpice", "EnforcePassivity:=" , False, "EnforceCausality:=" , False, "UseCommonGround:=" , True, "ShowGammaComments:=" , False, "Renormalize:=" , False, "RenormImpedance:=" , 50,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "FittingError:=" , 0.5,
        "MaxPoles:=" , 10000,
        "PassivityType:=" , "IteratedFittingOfPVLF",
        "ColumnFittingType:=" , "Matrix",
        "SSFittingType:=" , "FastFit",
        "RelativeErrorToleranc:=", False,
        "EnsureAccurateZfit:=" , True,
        "TouchstoneFormat:=" , "MA",
        "TouchstoneUnits:=" , "GHz",
        "TouchStonePrecision:=" , 15,
        "SubcircuitName:=" , "",
        "ExportDirectory:=" , "C:/Program Files/ANSYS Inc/v261/An-
        sysEM/Examples/Circuit/RF Microwave/Filters/",
        "ExportSpiceFileName:=" , "filter_layout_LinearFrequency__isolation_0_13_
        p1_8_70625mm_p2_8_61007mm_s1_0_414637mm_s2_1_40411mm_w1_0_443634mm_w2_0_
        479608mm_w50_1_00311mm_1_sp.sp",
        "FullwaveSpiceFileName:=", "filter_layout_LinearFrequency__isolation_0_
        13_p1_8_70625mm_p2_8_61007mm_s1_0_414637mm_s2_1_40411mm_w1_0_443634mm_w2_
        0_479608mm_w50_1_00311mm_1_sp.sp",
        "UseMultipleCores:=" , True,
        "NumberOfCores:=" , 10
])
```

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportMatlab

Writes the S-parameters to disk in Matlab format (*.m text file).

UI Access

From the **Network Data Explorer** tab, select **File** > **Save As** to open a **Save As** explorer window. Then select **MATLAB (*.m)** from the **Save as type:** drop-down menu.

Parameters

Name	Type	Description
<filePath>	String	The full path to the file.
<matrixType>	Integer	One of the following: • 0 – S-Parameters • 1 – Y-Parameters • 2 – Z-Parameters • 3 – Gamma • 4 – Impedance **Note:** The default value is **0**.
<formalType>	Integer	One of the following: • 0 – Magnitude/Angle (degrees) • 1 – Real/Imaginary • 2 – dB/Angle (degrees) **Note:** The default value is **0**.
<precision>	Integer	The number of significant figures. **Note:** The default value is **10**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<frequencies>	Array of Doubles	A subset of the calculated frequencies. Any array values that don't have calculated data will be ignored. An empty array will cause all frequencies to be written. **Note:** The default value is an empty array.
Return Value	Boolean True if file was successfully written; else False.		

Python Syntax	ExportMatlab()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.ExportMatlab("D:\folder\ne133.m", 0, 1, 12, [])`

ExportNMFData

Exports s-parameters in neutral file format (*.nmf).

UI Access	**Results** > **Solution Data**. On the **Matrix** tab, click **Export** tab. Then click **Export** > **S-parameter**. Select Neutral Model format and export.

Parameters

Name	Type	Description
<sourceName>	Array	String names of solutions. Use more than one array entry for a combined Interpolating sweep.
<outputFileName>	String	Full path of output file, including *.nmf extension.
<reduceMat>	String	Either "Original" or name of reduced matrix.
<refImp>	Double	Reference impedence.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<freqs>	Array	Either "All" or array of doubles representing frequencies.
	<variationKey>	String	Design variation key. Pass empty string for the current nominal variation.
	<displayFormat>	String	"MagPhase", "RealImag", "DbPhase".
	<pass>	Integer	Optional. The pass to export. This is ignored if the *sourceName* is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
	<ComplexFormat>	Integer	Optional. The format to use for the exported data. Options are 0 = Magnitude/Phase 1= Real/Imaginary. 2= db/Phase.
	<Precision>	Integer	Optional. Number of digits precision. If not specified, the default is 6.
Return Value	None.		

Python Syntax	ExportNMFData (<*sourceName*>, <*outputFileName*>,<*reduceMat*>, <*refImp*>, <*freqs*>,<*variationKey*>, <*displayFormat*>, <*pass*>, <*ComplexFormat*>, <*Precision*>)
Python Example	`oDesign.ExportNMFData("Setup2 : LastAdaptive", "C:/outputs/data.nmf, "Original", 50, [10000000, 20000000, 30000000, 40000000, 50000000, 60000000, 70000000, 80000000, 90000000, 100000000], "", "MagPhase", 0, 1, 11)`

ExportNetworkData

Exports matrix solution data to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<DesignVariationKey>	String	Design variation key. Pass empty string for the current nominal vari-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		ation.
<SolnSelectionArray>	Array	Array of selected solutions. `[<SolnSelector>,<SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating sweep.
<SolnSelector>	String	Solution setup name and solution name, separated by a colon.
<FileFormat>	Integer	File format value. 2 : Tab delimited spreadsheet format (.tab) 3 : Touchstone (.sNp) 4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<OutFile>	String	Full path to the file to write out.
<FreqsArray>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<DoRenorm>	Boolean	For Touchstone format only. Specifies whether to renormalize the data before export.
<RenormImped>	Double	For Touchstone format only. Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<DataType>	Array	Optional. Type: "S", "Y", or "Z". The matrix to export.
<pass>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<ComplexFormat>	Integer	Optional. Type: "0", "1", or "2"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.
	<Precision>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
	<UseExportFreqs>	Boolean	Specifies whether to use export frequencies.
	<IncludeGammaComments>	Boolean	Touchstone only. Specifies whether to include Gamma and Impedance comments.
	<SupportNonStdExport>	Boolean	Specifies whether to support non-standard Touchstone extensions for mixed reference impedance.
Return Value	None.		

Python Syntax	ExportNetworkData(*<DesignVariationKey>*, *<SolnSelectionArray>*, *<SolnSelector>*, *<FileFormat>*, *<OutFile>*, *<FreqsArray>*, *<DoRenorm>*, *<RenormImped>*, *<DataType>*, *<pass>*, *<ComplexFormat>*, *<Precision>*, *<UseExportFreqs>*, *<IncludeGammaComments>*, *<SupportNonStdExport>*)
Python Example	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportSpreadsheet

Writes the S-parameters to disk in spreadsheet format (*.tab text file).

<table>
<tr><td>UI Access</td><td colspan="3">From the Network Data Explorer tab, select File > Save As to open a Save As explorer window. Then select Data Table (spreadsheet) (*.tab) from the Save as type: drop-down menu.</td></tr>
<tr><td rowspan="4">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><filePath></td><td>String</td><td>The full path to the file.</td></tr>
<tr><td><matrixType></td><td>Integer</td><td>One of the following:
0 – S-Parameters
1 – Y-Parameters
2 – Z-Parameters
3 – Gamma
4 – Impedance
Note: The default value is 0.</td></tr>
<tr><td><formalType></td><td>Integer</td><td>One of the following:
0 – Magnitude/Angle (degrees)
1 – Real/Imaginary
2 – dB/Angle (degrees)
Note: The default value is 0.</td></tr>
<tr><td><precision></td><td>Integer</td><td>The number of significant figures.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: The default value is **10**.
	<frequencies>	Array of Doubles	A subset of the calculated frequencies. Any array values that don't have calculated data will be ignored. An empty array will cause all frequencies to be written. **Note:** The default value is an empty array.
	<renormalize>	Boolean	If **True**, will renormalize the data to the value of renormImpedance before writing. **Note:** The default value is **False**.
	<renormImpedance>	Double	Data will be renormalized to this impedance before writing only if renormalize is **True**. **Note:** The default value is **50 ohms**.
Return Value	Boolean True if file was successfully written; else False.		

Python Syntax	ExportSpreadsheet()

Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`
	`oData = oNDE.Open(<model path>)`
	`success = oData.ExportSpreadsheet("D:\folder\ne133.tab", 0, 1, 12, [], True, 23)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

ExportTouchstone

Writes the S-parameters to disk in Touchstone 1.0 format (*.s*n*p text file).

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select File > Save As to open a Save As explorer window. Then select Touchstone Format 1.0 (*.snp) from the Save as type: drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><filePath></td><td>String</td><td>The full path to the file.</td></tr>
<tr><td><matrixType></td><td>Integer</td><td>One of the following:
0 – S-Parameters
1 – Y-Parameters
2 – Z-Parameters
3 – Gamma
4 – Impedance
Note: The default value is 0.</td></tr>
<tr><td><formalType></td><td>Integer</td><td>One of the following:
0 – Magnitude/Angle (degrees)
1 – Real/Imaginary
2 – dB/Angle (degrees)</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: The default value is **0**.
	<precision>	Integer	The number of significant figures. **Note:** The default value is **10**.
	<frequencies>	Array of Doubles	A subset of the calculated frequencies. Any array values that don't have calculated data will be ignored. An empty array will cause all frequencies to be written. **Note:** The default value is an empty array.
	<renormalize>	Boolean	If **True**, will renormalize the data to the value of renormImpedance before writing. **Note:** The default value is **False**.
	<renormImpedance>	Double	Data will be renormalized to this impedance before writing only if renormalize is **True**. **Note:** The default value is **50 ohms**.
	<freqUnits>	String	One of the following: "Hz", "KHz", "MHz", "GHz". **Note:** The default value is **"Hz"**.
Return Value	Boolean True if file was successfully written; else False.		

Python	ExportTouchstone()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax	
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.ExportTouchstone("D:\folder\ne133.s2p", 0, 1, 12, [], True, 23, "GHz")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

ExportTouchstone2

Writes the S-parameters to disk in Touchstone 2 format (*.ts text file).

UI Access	From the **Network Data Explorer** tab, select **File** > **Save As** to open a **Save As** explorer window. Then select **Touchstone 2 Format (*.ts)** from the **Save as type:** drop-down menu.
Parameters	See table below

Name	Type	Description
<filePath>	String	The full path to the file.
<matrixType>	Integer	One of the following: • 0 – S-Parameters • 1 – Y-Parameters • 2 – Z-Parameters • 3 – Gamma • 4 – Impedance **Note:** The default value is **0**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<formalType>	Integer	One of the following: • 0 – Magnitude/Angle (degrees) • 1 – Real/Imaginary • 2 – dB/Angle (degrees) **Note:** The default value is **0**.
<precision>	Integer	The number of significant figures. **Note:** The default value is **10**.
<frequencies>	Array of Doubles	A subset of the calculated frequencies. Any array values that don't have calculated data will be ignored. An empty array will cause all frequencies to be written. **Note:** The default value is an empty array.
<renormalize>	Boolean	If **True**, will renormalize the data to the value of renormImpedance before writing. **Note:** The default value is **False**.
<renormImpedance>	Double	Data will be renormalized to this impedance before writing only if renormalize is **True**. **Note:** The default value is **50 ohms**.
<freqUnits>	String	One of the following: "Hz", "KHz", "MHz", "GHz". **Note:** The default value is **"Hz"**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Boolean True if file was successfully written; else False.

Python Syntax	ExportTouchstone2()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.ExportTouchstone2("D:\folder\ne133.ts", 0, 1, 12, [], True, 23, "GHz")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Extract (SPISim)

Creates a new smaller network data after removing data for the specified terminal port numbers.

Note: Extract opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Extract** from the drop-down menu.
Parameters	

Name	Type	Description
<oData>	Object	Data from this object is copied to a new network data and the copy is transformed.
<portNumbers>	Array of Integers	The port numbers that will be retained (integers between 1 and *n*).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Network IDispatch.

Python Syntax	Extract()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Extract(oData1, [3, 2])`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetFrequencies

Returns the frequency values with calculated data in the network data.

UI Access	None.
Parameters	None.
Return Value	Array of doubles.

Python Syntax	GetFrequencies()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `Freqs = oData.GetFrequencies()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetFrequencyCount

Returns the number of frequencies with calculated data in the network data.

UI Access	None.
Parameters	None.
Return Value	Integer number of frequencies.

Python Syntax	GetFrequencyCount()
Python Example	`oData = oDesktop.GetTool("ndExplorer")` `numFreqs = oData.GetFrequencyCount()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetName

Returns the name of the network data.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	None.
Parameters	None.
Return Value	String name.

Python Syntax	GetName()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `strName = oData.GetName()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetPortCount

Returns the total number of ports.

UI Access	From the **Network Data Explorer** tab, select **Edit Ports** in the **NDE** ribbon to open the **Port properties** window.
Parameters	None.
Return Value	Integer number of ports.

Python Syntax	GetPortCount()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`num = oData.GetPortCount()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetPortNumber

Returns the terminal port number for a given terminal port name.

Note: GetPortNumber can be used in other commands that require a port number, if a port name is preferable.

The PortNumber is the index of the name port in the original data and is not changed by any other settings applied to the oPostProc, including the Reorder settings. It is permanently linked to the port name and can be used interchangeably with the port name in many of the PostProcSettings script functions.

UI Access	From the **Network Data Explorer** tab, ensure the **Full Port Names** check box is activated. Port numbers will be represented in the object portNames.
Parameters	**Name** / **Type** / **Description** *<portName>* / String / The designation of the port number.
Return Value	Integer.

Python Syntax	GetPortNumber()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`index = oPostProc.GetPortNumber("Input35")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetPostProcSettings

Returns a copy of the IDispatch to the postprocessing settings for the network data (oPostProc).

Note: Making changes to the PostProcSettings will not change the network data. To make changes, call oData.SetPostProcSettings to change the network data.

UI Access	None.
Parameters	None.
Return Value	PostProcSettings IDispatch

Python Syntax	GetPostProcSettings()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc = oData.GetPostProcSettings()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetSolutionVariation

Returns the network data object corresponding to a variation of the solution.

UI Access	None.
Parameters	**Name** / **Type** / **Description** *<solutionID>* / Integer / ID of the solution (obtained with LoadSolution). *<variation>* / String / The variation key.
Return Value	Network IDispatch.

Python Syntax	GetSolutionVariation()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData = oNDE.GetSolutionVariation("O, "offset='0.1'")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

GetVariation

Returns the variation key for the network data.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	None.
Parameters	None.
Return Value	String variation key.

Python Syntax	GetVariation()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `variation = oData.GetVariation()`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

HasSameData

Compares one network data object with another network data. Actual calculated values will be compared and no interpolation will be done.

UI Access	None.		
Parameters	**Name**	**Type**	**Description**
	<NetworkData>	Object	The second network data to compare against.
	<matrixType>	Integer	One of the following: • 0 – S-Parameters • 1 – Y-Parameters • 2 – Z-Parameters

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• 3 – Gamma • 4 – Impedance **Note:** The default value is **0**.
	<comparePortNames>	Boolean	If **True**, port names must be identical. **Note:** The default value is **True**.
	<compareNoise>	Boolean	If **True**, and both network data have noise data, the comparison will fail unless the noise values also match. **Note:** The default value is **10**.
	<relativeTolerance>	Double	If the absoluteTolerance test fails, then abs(value1 - value2)/max (abs(value1), abs(value2)) must be less than this to be considered equal. **Note:** The default value is **1e-14**.
	<absoluteTolerance>	Double	If **True**, then abs(value1 - value2) < absoluteTolerance. **Note:** The default value is **0**.
Return Value	Boolean True if the compared network data have the same frequency and matrix values; otherwise, False.		

Python Syntax	HasSameData()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.HasSameData(oData2, 0, True, False, 1e-10, 0)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

LoadSolution

Loads the specified design (all variations) and returns an integer ID.

UI Access	After analyzing a design, from the **Project Manager** window, expand the **Project Tree** and Analysis folders. Then right-click on the completed analysis icon and select **Network Data Explorer** to open the solution in the **Network Data Explorer** tab.
Parameters	**Name** / **Type** / **Description** *<projectName>* / String / Full path to the Electronics Desktop file. *<designName>* / String / Name of the design. *<solutionName>* / String / Name of the solution.
Return Value	Integer ID (identifying the solution); < 0 if the solution is not loaded.

Python Syntax	LoadSolution()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `id = oNDE.LoadSolution("D:\folder\Tee.aedt", "HFSS_Test", "Setup1:Sweep1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Open

Reads contents of file and returns the IDispatch for the new network data.

UI Access	From the **Network Data Explorer** tab, select **Open** in the **NDE** ribbon, or select **File** > **Open** to open an explorer window. Then navigate to the required S-parameter file.
Parameters	**Name** / **Type** / **Description** *<fileName>* / String / Full path to the file containing S-parameters.
Return Value	Network IDispatch.

Python Syntax	Open()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData = oNDE.Open("D:\folder\test.s2p")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Rename (SPISim)

Creates a new network data of the same size with the terminal ports renamed.

> **Note:** Rename opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Rename** from the drop-down menu.
Parameters	**Name** / **Type** / **Description** *<oData>* / Object / Data from this object is copied to a new network data and the copy is transformed. *<portNames>* / Array of Strings / The new names (one for each port).
Return Value	Network IDispatch.

Python Syntax	Rename()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Rename(oData1, ["input", "control", "output"])`

> **Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Renormalize (SPISim)

Creates a new network data of the same size after renormalizing the terminal ports using the specified impedance values.

> **Note:** Rename opens and interacts with **SPISim** to complete.

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select Transforms in the NDE ribbon. Then select Renormalize from the drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><oData></td><td>Object</td><td>Data from this object is copied to a new network data and the copy is transformed.</td></tr>
<tr><td><portImpedances></td><td>Array of Doubles</td><td>The new impedance values to be applied to each port.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Network IDispatch.

Note: Renormalize returns None if there is not an impedance value for each port.</td></tr>
</table>

Python Syntax	Renormalize()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Renormalize(oData1, [3, 2])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Reorder

Rearranges the terminal port entries but does not change the port names (i.e., switching first and third terminal port positions will give switched values for S(1,1) and S(3,3) but S(Port1,Port1) and S(Port3,Port3) do not change).

UI Access	From the **Network Data Explorer** tab, select **Edit Ports** in the **NDE** ribbon to open the **Port properties** window. **Click+drag** the terminal port rows to reorder them. Make changes, as necessary, then click **OK**.
Parameters	**Name** / **Type** / **Description** *<newOrder>* / Array of Integers / Must have one entry for each port; all numbers 1 to *n* must be present.
Return Value	None.

Python Syntax	Reorder()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.Reorder([3, 2, 1])`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Reorder (SPISim)

Creates a new network data with the same number of terminal ports and with the port names in the same order. The data is also reordered so it corresponds to the new order (e.g., the data for S(Port1, Port1) may correspond to S(Port3, Port3).

> **Note:** Reorder opens and interacts with **SPISim** to complete.

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select Transforms in the NDE ribbon. Then select Reorder from the drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><oData></td><td>Object</td><td>Data from this object is copied to a new network data and the copy is transformed.</td></tr>
<tr><td><portNumbers></td><td>Array of Integers</td><td>The port numbers in new order (integers between 1 and n).</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Network IDispatch.

Note: Reorder returns None if the array argument does not contain all the terminal port numbers, in any order.</td></tr>
</table>

Python Syntax	Reorder()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Reorder([3, 2, 1])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Reset

Resets the post-processing to what it was when the network data was created.

Note: This is not equivalent to setting post-processing to an empty state. The network data may have been read from a file or created from solution data that already had post-processing settings.

Resetting the oPostProcSetting object does not change any data; it only changes the settings object. If you want to change a data object to apply the changed settings, you must follow the Reset function call with oData.SetPostProcSettings(oPostProc)

UI Access	From the **Network Data Explorer** tab, select **Reset postprocessing** in the **NDE** ribbon.
Parameters	None.
Return Value	None.

Python Syntax	Reset()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.Reset()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

SetAllPortImpedances

Sets all terminal port impedances in a single call. The impedances are in an array of real or complex values.

UI Access	Through the UI, the terminal ports can **only** be edited individually. From the **Network Data Explorer** tab, select **Edit Ports** in the **NDE** ribbon to open the **Port properties** window. Make changes, then click **OK**.
Parameters	**Name** / **Type** / **Description** *<impedances>* / Array of Doubles or String / Must have one entry for each port or a single complex (or real) value which will be applied to all ports.
Return Value	None.

Python Syntax	SetAllPortImpedances()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.SetAllPortImpedances([23, "2+3i", 50])` **`-or-`** `oData.SetAllPortImpedances(50)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetPortDeembedDistance

Sets terminal port impedance for a single port.

UI Access	From the **Network Data Explorer** tab, select **Edit Ports** in the **NDE** ribbon to open the **Port properties** window. Make changes, then click **OK**.
Parameters	**Name** / **Type** / **Description** *<portNumber>* / Integer / The port number of the changed terminal port (integers between 1 and **n**) . *<distance>* / String / Impedance with units (e.g., “20mm”); if no units, meters are assumed
Return Value	None.

Python Syntax	SetPortDeembedDistance()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.SetPortDeembedDistance(2, "20mm")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

SetPortImpedance

Sets port impedance for a single port.

UI Access	From the **Network Data Explorer** tab, select **Edit Ports** in the **NDE** ribbon to open the **Port properties** win-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	dow. Make changes, then click **OK**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*portNumber*></td><td>Integer</td><td>The port number of the changed terminal port (integers between 1 and *n*).</td></tr><tr><td><*impedance*></td><td>Double or String</td><td>Double if impedance value is real; string format (e.g., 24+10i) for complex impedance.</td></tr></table>
Return Value	None.

Python Syntax	SetPortImpedance()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.SetPortImpedance(2, "25+3i")`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

SetPostProcSettings

Applies postprocessing settings (oPostProc) to the specified network data.

Note: Making changes to the PostProcSettings will not change the network data. To make changes, call oData.SetPostProcSettings to change the network data.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	None.
Parameters	**Name** / **Type** / **Description** *<PostProcSettings>* / Object / The settings that will be applied to the data.
Return Value	None.

Python Syntax	SetPostProcSettings()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData.SetPostProcSettings(oPostProc)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Smooth

Creates a new network data with values smoothed. The number of adjacent points smoothed is user-specified.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Smooth** from the drop-down menu.
Parameters	**Name** / **Type** / **Description** *<oData>* / Object / Data from this object is copied to a new network data and the copy is transformed. *<order>* / Integer / The number of adjacent points that are smoothed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: The default value is **10**.
	<enforceCausality>	Boolean	Casual data is calculated before smoothing. **Note:** The default value is **False**.
Return Value	Network IDispatch **Note:** Network IDispatch is returned only if the Smooth function was successfully able to perform the smoothing operation on the supplied data and meet the maximum 0.025 error tolerance. Otherwise it will return a < null object >. The user may want to try changing other input parameters, for example, `enforceCausality`.		

Python Syntax	Smooth()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Smooth(oData1, 3, True)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Stretch (SPISim)

Creates a new network data representing a matrix of transmission lines with the length of those transmission lines multiplied by a factor of *n*.

> **Note:** Stretch opens and interacts with **SPISim** to complete.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Stretch** from the drop-down menu.
Parameters	**Name** / **Type** / **Description** *<oData>* / Object / Data from this object is copied to a new network data and the copy is transformed. *<factor>* / Double / The transmission line lengths are multiplied by this number.
Return Value	Network IDispatch. **Note:** Stretch returns **None** if the network data argument does not represent a matrix of transmission lines.

Python Syntax	Stretch()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Stretch(oData1, 3.1)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

Terminate

Creates a new network data with specified ports terminated.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Terminate** from the drop-down menu.
Parameters	**Name** / **Type** / **Description** *<oData>* / Object / Data from this object is copied to a new network data and the copy is transformed. *<portNumbers>* / Array of Integers / The port numbers that are terminated (integers between 1 and *n*). *<termImpedances>* / Array of Complex Numbers / The impedance values used to terminate the specified ports.
Return Value	Network IDispatch. **Note:** Terminate returns **None** if the port numbers are not valid or there are no impedance value for the port numbers.

Python Syntax	Terminate()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oData2 = oNDE.Terminate(oData1, [2, 3], [23, "10+2i"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

20 - CompInstance Script Commands

CompInstance commands should be executed by the **oDesign2** or **oPropHost2** object.

```
oDesign2 = CompInstance.GetParentDesign
```

```
oPropHost2 = CompInstance.GetPropHost
```

Callback Scripting Using CompInstance Object

Callback scripts are scripts that can be set in the Property Dialog for individual properties by clicking the button in the Callback column and choosing a script that is saved with the project. Callback scripts can contain any legal script commands including general Ansys script function calls (e.g., GetApplicationName()). In addition, Callback scripts can also call functions on a special object named CompInstance.

You can obtain an interface to a CompInstance in a schematic or layout by calling oEditor.GetCompInstanceFromRefDes(refDes). For more information see Layout Scripting and Schematic Scripting. This interface is also available as a CompInstance object in CompInstance event callbacks, such as placing a component in a layout or schematic.

Definitions

<**propName**> = text string

<**value**> = double

<**valueText**> = text string

<**fileName**> = full path file name

<**choices**> = string containing menu choices separated by commas

<**initialChoice**> = string containing initial choice for menu; must be one of the <choices>

<**scriptName**> = string containing name of script stored in project

<**bool**> is 1 for true or 0 for false

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<**editorName**> is either "Layout" or "SchematicEditor"

The topics for this section include:

CompInstance Functions

CompInstance Functions

Following are commands that can be used to manipulate properties from a CompInstance script.

The topics for this section include:

GetComponentName

GetEditor

GetInstanceID

GetInstanceName

GetParentDesign

GetPropHost

GetPropServerName

GetComponentName

Returns the name of the component corresponding to this CompInstance.

UI Access	NA		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	None		
Return Value	String name of component (e.g. MS_TRL) and stores it in "name"		

Python Syntax	GetComponentName()
Python Example	`name = CompInstance.GetComponentName()`

GetInstanceID [Component Instance]

Use: Returns the instanceID of the CompInstance.

Command: None

Syntax: GetInstanceID()

Return Value: String

Python Syntax	GetInstanceID()
Python Example	`id = CompInstance.GetInstanceID()`

GetInstanceName [Component Instance]

Use: Returns the instance name of the component corresponding to this CompInstance.

Command: None

Syntax: GetInstanceName()

Return Value: String

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Returns instanceName (e.g. A7) of compInstance and stores it in "name".

Note that the Instance Name is not the same as the RefDes.

GetParentDesign

Use: Returns an interface to the compInstance's parent design. This interface can be used to call Design functions. See: Design Object Script Commands.

Command: None

Syntax: GetParentDesign()

Return Value: Returns interface to design.

Python Syntax	GetParentDesign()
Python Example	`oDesign2 = CompInstance.GetParentDesign()`

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropServerName

Returns the PropServerName of the Component corresponding to this CompInstance.

UI Access	Not available
Parameters	None
Return Value	PropServerName in the form of a string

Python Syntax	GetPropServerName()
Python Example	`name = CompInstance.GetPropServerName()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

21 - Simulation Setup Commands

This section presents the commands that are available in the SolveSetups module. These commands should be executed by the **oDesign** object. For example:

```
oDesign.GetModule("SolveSetups")

oModule.CommandName <args>
```

The following commands are described in this section:

Add (Layout Editor)

AddSimulationSetup [Nexxim]

AddMeshOperation

AddSweep

AnalyzeSweep

Delete

DeleteGFD

DeleteSweep

DynamicMeshOverlays

Edit

EditSweep

ExportToHFSS

ExportToQ3D

GenerateMesh

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAllSolutionNames

GetGFDData

GetMeshOperations

GetSetupData

GetSetups

GetSweeps

LayoutMeshOverlay

Layout3DMeshOverlay

ListVariations

RefreshMeshOverlays

Rename

RenameSimulationSetup

RenameSweep

Validate

Add (Layout Editor)

Adds a new setup.

> **Note:** The following tables display **example** parameters applicable to a Near Field solution setup. Selecting a different setup will also add, remove, or change the applicable parameters.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>From HFSS 3D Layout, choose Solution Setup and then select from the available solution setups.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SetupArray></td><td>Array</td><td>An array that contains these parameters in order:
<NAME:SetupName>
<NameArray>
<CustomSetup>
<Position>
<SimSetupType>
<SimSettingsArray>
<SolveSetupType>
This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.</td></tr>
<tr><td><NAME:SetupName></td><td>String</td><td>User-entered name of the new setup.</td></tr>
<tr><td><NameArray></td><td>Array</td><td>An array that contains these parameters in order:
<NAME:Properties>
<Enable></td></tr>
<tr><td><NAME:Properties></td><td>String</td><td>Identifies the contents of the array (i.e., Properties).</td></tr>
<tr><td><Enable></td><td>Boolean</td><td>True – setup is active
False – setup is inactive</td></tr>
<tr><td><SimSetupType></td><td>String</td><td>Identifies the type of setup (e.g., kSIwaveNearField).</td></tr>
<tr><td><SimSettingArray></td><td>Array</td><td>Optional. An array that contains these parameters in order:
<NAME:SimulationSettings>
<Enabled>
<UseSISettings>
<UseCustomSettings>
<SISliderPros>
<PISliderPros></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <SIWAdvancedArray> • <SIWDCArray> • <SIWDCAdvancedArray> • <SIWSParamArray> • <NearFieldSettingArray> • <NAME:SweepDataList> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SimulationSettings>	String	Identifies the contents of the array (i.e., SimulationSettings).
<Enabled>	Boolean	**True** – array settings are active **False** – array settings are inactive
<UseSISettings>	Boolean	**True** – uses SI Simulation settings **False** – uses PI Simulation settings
<UseCustomSettings>	Boolean	**True** – overrides <UseSISettings> and activates <SIWAdvancedArray> parameters **False** groups by range
<SISliderPros>	Integer	**0** – optimizes simulation for speed **1** – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
<PISliderPros>	Integer	**0** – optimizes simulation for speed **1** – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
<SIWAdvancedArray>	Array	*Optional*. An array that contains these parameters in order: • <NAME:SIWAdvancedSettings> • <IncludeCoPlaneCoupling> • <IncludeInterPlaneCoupling> • <IncludeSplitPlaneCoupling>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <IncludeFringeCoupling> • <IncludeTraceCoupling> • <XtalkThreshold> • <MaxCoupledLines> • <MinVoidArea> • <MinPadAreaToMesh> • SnapLengthThreshold> • <MeshAutoMatic> • <MeshFrequency> • <ReturnCurrentDistribution> • <INcludeVISources> • <IncludeInfGnd> • <InfGndLocation> • <PerformERC> • <IgnoreNonFunctionalPads> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<NAME:SIWAdvancedSettings>	String	Identifies the contents of the array (i.e., SIWAdvancedSettings).
	<IncludeCoPlaneCoupling>	Boolean	**True** – coplane coupling active **False** – coplane coupling inactive
	<IncludeInterPlaneCoupling>	Boolean	**True** – intra-plane coupling active **False** – intra-plane coupling inactive
	<IncludeSplitPlaneCoupling>	Boolean	**True** – split-plane coupling active **False** – split-plane coupling inactive
	<IncludeFringeCoupling>	Boolean	**True** – cavity field coupling active **False** – cavity field coupling inactive
	<IncludeTraceCoupling>	Boolean	**True** – trace coupling active **False** – trace coupling inactive

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<XtalkThreshold>	Integer	Maximum cross-talk threshold, in decibels.
<MaxCoupledLines>	Integer	The number of allowed coupled lines.
<MinVoidArea>	Integer	Minimum size of voids; voids larger than the minimum are meshed.
<MinPadAreaToMesh>	Integer	Minimum size of pads; pads larger than the minimum are meshed.
SnapLengthThreshold>	Integer	Minimum distance of vertices; vertices less than the minimum are snapped.
<MeshAutoMatic>	Boolean	**True** – mesh are automatically refined **False** – mesh are not automatically refined
<MeshFrequency>	Integer	Specific frequency at which mesh are refined.
<ReturnCurrentDistribution>	Boolean	**True** – returns current distribution is traced **False** – returns current distribution is not traced
<IncludeVISources>	Boolean	**True** – parasitic elements from voltage/current sources are included **False** – parasitic elements are not included
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
<InfGndLocation>	Integer	Distance below bottom layer in stackup to place infinite ground plane.
<PerformERC>	Boolean	**True** – performs ERC during simulation **False** – does not perform ERC during simulation
<IgnoreNonFunctionalPads>	Boolean	**True** – excludes non-functional pads from resonant mode, frequency sweep, and SYZ simulations **False** – does not exclude non-functional pads from resonant mode, frequency sweep, and SYZ simulations.
<SIWSParamSettings>	Array	*Optional*. An array that contains these parameters in order: • <UseStateSpace> • <Interpolation> • <Extrapolation>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <DCBehavior> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<UseStateSpace>	Boolean	**True** – uses the **State Space Model** **False** – uses a **Custom Model**
<Interpolation>	Integer	**0** – sets **Interpolation** to **Point** **1** – sets **Interpolation** to **Point** **2** – sets **Interpolation** to **Point**
<Extrapolation>	Integer	**0** – sets **Extrapolation** to **Zero padding** **1** – sets **Extrapolation** to **Same as last point** **2** – sets **Extrapolation** to **Linear extrapolation from last 2 points** **3** – sets **Extrapolation** to **Constant magnitude linear phase extrapolation**
<DCBehavior>	Integer	**0** – sets **DC Behavior** to **Zero padding** **1** – sets **DC Behavior** to **Same as last point** **2** – sets **DC Behavior** to **Linear extrapolation from last 2 points** **3** – sets **DC Behavior** to **Constant magnitude linear phase extrapolation** **4** – sets **DC Behavior** to **Series Capacitor** **5** – sets **DC Behavior** to **Leave all signal lines open circuited**
<SIWSNearFieldArray>	Array	An array that contains these parameters in order: • <NAME:SIWNearFieldSettings> • <UseExternalSources> • <ExternalSourceFilePath> • <InterpolateSpectrum> • <IncludeInfGnd> • <PosXoffset> • <NegXoffset>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <PosYoffset> • <NegYoffset> • <PosZoffset> • <NegZoffset> • <MinAdaptPasses> • <MaxAdaptPasses> • <NewTrianglesPerPass> • <GlobalErrorTol> • <UseUserEdgeLength> • <MeshEdgeLength> • <ExportData> • <ExportDataPath> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SIWNearFieldSettings>	String	Identifies the contents of the array (i.e., SIWNearFieldSettings).
<UseExternalSources>	Boolean	**True** – uses external **Excitations** definitions **False** – uses **Excitations** definitions defined in project
<ExternalSourceFilePath>	String	Directory path to external **Excitations** definitions file.
<InterpolateSpectrum>	Boolean	**True** – interpolates spectrum at missing frequency points (external **Excitations** definitions file only) **False** – does not interpolate spectrum at missing frequency points
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
PosXoffset>	Integer	Value determines the +X offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegXoffset>	Integer	Value determines the -X offset of a cube that encloses the

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		design; the Near Field is evaluated on the surface of this cuboid.
<PosYoffset>	Integer	Value determines the +Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegYoffset>	Integer	Value determines the -Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<PosZoffset>	Integer	Value determines the +Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegZoffset>	Integer	Value determines the -Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<MinAdaptPasses>	Integer	Minimum number of passes the adaptive analysis will make, even if convergence criteria are met.
<MaxAdaptPasses>	Integer	Maximum number of passes the adaptive analysis will make, unless convergence criteria are met.
<NewTrianglesPerPass>	Integer	Percentage of triangles to add per pass.
<GlobalErrorTol>	Integer	Maximum relative number of errors to allow between two successive adaptive passes.
<UseUserEdgeLength>	Boolean	**True** – uses custom maximum edge length **False** – automatically determines maximum edge length
<MeshEdgeLength>	Integer	Maximum edge length.
<ExportData>	Boolean	**True** – exports Near Field data **False** – does not export Near Field data
<ExportDataPath>	String	Path to directory in which to save Near Field data.
<SIWTemperatureArray>	Array	An array that contains these parameters in order: • <NAME:SIWTemperatureSettings> • <UseSitempFile> • <SitempFilePath> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Note: The following tables display **example** parameters applicable to a Near Field solution setup. Selecting a different setup will also add, remove, or change the applicable parameters.
	<NAME:SIWTemperatureSettings>	String	Identifies the contents of the array (i.e.,SIwave temperature settings).
	<UseSitempFile>	Boolean	**True** – uses external Icepak simulation results or temperature map **False** – does not use external temperature map
	<SitempFilePath>	String	Directory path to external Icepak temperature map file.
	<NAME:SweepDataList>	String	Identifies the determining factors of the Near Field simulation (i.e., sweeps) (i.e., SweepDataList).
	<SolveSetupType>	String	Identifies the type of setup (i.e., SIwaveNearField).
Return Value	None		

Python Syntax	oModule.Add([<NAME:SIwaveNearField1>, [<NAME:Properties>, <Enable>], <SimSetupType>, [<NAME:SimulationSettings>, <Enabled>, <UseSISettings>, <UseCustomSettings>, <SISliderPos>, <PISliderPos>], [<NAME:SIWAdvancedSettings>, <IncludeCoPlaneCoupling>, <IncludeInterPlaneCoupling>, <IncludeSplitPlaneCoupling>, <IncludeFringeCoupling>, <IncludeTraceCoupling>, <XtalkThreshold>, <MaxCoupledLines>, <MinVoidArea>, <MinPadAreaToMesh>, <SnapLengthThreshold>, <MeshAutoMatic>, <MeshFrequency>, <ReturnCurrentDistribution>, <IncludeVISources>, <IncludeInfGnd>, <InfGndLocation>, <PerformERC>, <IgnoreNonFunctionalPads>], [<NAME:SIWSParamSettings>, <UseStateSpace>, <Interpolation>, <Extrapolation>, <DCBehavior>], [<NAME:SIWNearFieldSettings>, <UseExternalSources>, <ExternalSourceFilePath>, <InterpolateSpectrum>, <IncludeInfGnd>, <PosXoffset>, <NegXoffset>, <PosYoffset>, <NegYoffset>, <PosZoffset>, <NegZoffset>, <MinAdaptPasses>, <MaxAdaptPasses>, <NewTrianglesPerPass>, <GlobalErrorTol>, <UseUserEdgeLength>, <MeshEdgeLength>, <ExportData>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ExportDataPath>] [<NAME:SIWTemperatureSettings>, <UseSitempFile>, <SitempFilePath>]], [<NAME:SweepDataList>], <SolveSetupType>])
Python Example	oModule.Add(["NAME:SIwaveNearField1", ["NAME:Properties", "Enable:=" , "true"], "SimSetupType:=" , "kSIwaveNearField", ["NAME:SimulationSettings", "Enabled:=" , True, "UseSISettings:=" , False, "UseCustomSettings:=" , False, "SISliderPos:=" , 2, "PISliderPos:=" , 0, ["NAME:SIWAdvancedSettings", "IncludeCoPlaneCoupling:=", False, "IncludeInterPlaneCoupling:=", False, "IncludeSplitPlaneCoupling:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "IncludeFringeCoupling:=", False,
        "IncludeTraceCoupling:=", False,
        "XtalkThreshold:="      , "-34",
        "MaxCoupledLines:="     , 12,
        "MinVoidArea:="         , "2mm2",
        "MinPadAreaToMesh:="    , "1mm2",
        "MinPlaneAreaToMesh:="  , "6.25e-6mm2",
        "SnapLengthThreshold:=" , "2.5um",
        "MeshAutoMatic:="       , True,
        "MeshFrequency:="       , "4GHz",
        "ReturnCurrentDistribution:=", False,
        "IncludeVISources:="    , False,
        "IncludeInfGnd:="       , False,
        "InfGndLocation:="      , "0mm",
        "PerformERC:="          , False,
        "IgnoreNonFunctionalPads:=", True
      ],
      [
        "NAME:SIWSParamSettings",
        "UseStateSpace:="       , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Interpolation:="       , 1,
        "Extrapolation:="       , 3,
        "DCBehavior:="          , 3
      ],
      [
        "NAME:SIWNearFieldSettings",
        "UseExternalSources:="  , False,
        "ExternalSourceFilePath:=", "",
        "InterpolateSpectrum:=" , True,
        "IncludeInfGnd:="       , False,
        "PosXoffset:="          , "1mm",
        "NegXoffset:="          , "1mm",
        "PosYoffset:="          , "1mm",
        "NegYoffset:="          , "1mm",
        "PosZoffset:="          , "1mm",
        "NegZoffset:="          , "1mm",
        "MinAdaptPasses:="      , 1,
        "MaxAdaptPasses:="      , 10,
        "NewTrianglesPerPass:=" , 20,
        "GlobalErrorTol:="      , 3,
        "UseUserEdgeLength:="   , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "MeshEdgeLength:="      , "1mm",
          "ExportData:="          , False,
          "ExportDataPath:="      , ""
        ],
          "NAME:SIWTemperatureSettings",
          "UseSitempFile:="       , False,
          "ExternalSourceFilePath:=", "",
        ],
        [
          "NAME:SweepDataList"
        ],
        "SolveSetupType:="      , "SIwaveNearField"
      ])
```

AddMeshOperation

Adds a mesh operation to a specified solution.

UI Access	From the **Project Manager**, right-click a **solution** and select **Assign Mesh Operation** > [**Any Submenu Item**].
Parameters	

Name	Type	Description
<name>	String	The name of the solution for which you want to apply a mesh operation.
<MeshArray>	Array	Structured array:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		["NAME:<mesh operation name>", "RefineInside:=" , <boolean True for inside; False for on surface>, "Type:=" , <string "LengthBased" or "SkinDepthLengthBased">, "Enabled:=" , <boolean True to enable operation; False to disable>, ["NAME:Assignment", <MeshEntityInfo>, <MeshEntityInfo>, <MeshEntityInfo>,...] "Region:=" , <string name of region, if selecting one>, "RestrictElem:=" , <boolean True to restrict number of mesh elements>, "NumMaxElem:=" , <string containing maximum number of mesh elements>, "RestrictLength:=", <boolean True to restrict length of mesh elements>, "MaxLength:=" , <string containing maximum length of mesh element>]
<MeshEntityInfo>	Array	Structured array. Add one MeshEntityInfo array for each entity selected. ["NAME:MeshEntityInfo", "IsFcSel:=" , <boolean>, "EntID:=" , <integer>, "FcIDs:=" , <array>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:MeshBody", "Id:=" , <integer>, "Nam:=" , <string>, "Layer:=" , <string layer name>, "Sheet:=" , <boolean>, "Net:=" , <string net name or "<no-net>">, "OrigNet:=" , <string net name or "<no-net>">], "BBox:=" , <array>]
Return Value	None.		

Python Syntax	AddMeshOperation(<*SolutionName*>, <*MeshArray*>)
Python Example	oModule.AddMeshOperation('HFSS Solution 1', ["NAME:Length 2", "RefineInside:=" , False, "Type:=" , "LengthBased", "Enabled:=" , True,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
"NAME:Assignment",
  [
  "NAME:MeshEntityInfo",
  "IsFcSel:="     , False,
  "EntID:="       , -1,
  "FcIDs:="       , [],
    [
    "NAME:MeshBody",
    "Id:="          , -1,
    "Nam:="         , "",
    "Layer:="       , "UNNAMED_  0",
    "Sheet:="       , True,
    "Net:="         , "<no-net>",
    "OrigNet:="     , "<no-net>"
    ],
  "BBox:="        , []
  ],
  [
  "NAME:MeshEntityInfo",
  "IsFcSel:="     , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "EntID:="       , -1,
      "FcIDs:="       , [],
        [
        "NAME:MeshBody",
        "Id:="          , -1,
        "Nam:="         , "",
        "Layer:="       , "UNNAMED_  0",
        "Sheet:="       , True,
        "Net:="         , "PCIE_VSS_C",
        "OrigNet:="     , "PCIE_VSS_C"
        ],
      "BBox:="        , []
      ],
    ],
  "Region:="        , "EXTENTS:rect_1076",
  "RestrictElem:="  , False,
  "NumMaxElem:="    , "1000",
  "RestrictLength:=", True,
  "MaxLength:="     , "0.2mm"
  ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	)

AddSweep (Layout Editor)

Adds a sweep to a setup.

UI Access	From the **Project Manager** window, right-click a setup and select **Add Frequency Sweep**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><Parameters></td><td>Array</td><td>A structured array containing the name and parameters for the new frequency sweep.</td></tr></table>
Return Value	None

Python Syntax	oModule.AddSweep(<SetupName>, [<Parameters>])

Python Example	<pre>oModule.AddSweep("Q3D1",
	  [
	  "NAME:Sweep1",
	  [
	  "NAME:Properties",
	  "Enable:=" , "true"
	  ],
	  [</pre>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:Sweeps",
        "Variable:=" , "Sweep1",
        "Data:=" , "LIN 10MHz 100MHz 10MHz",
        "OffsetF1:=" , False,
        "Synchronize:=" , 0
      ],
      "GenerateSurfaceCurrent:=", False,
      "SaveRadFieldsOnly:=" , False,
      "ZoSelected:=" , False,
      "SAbsError:=" , 0.005,
      "ZoPercentError:=" , 1,
      "GenerateStateSpace:=" , False,
      "EnforcePassivity:=" , False,
      "PassivityTolerance:=" , 0.0001,
      "UseQ3DForDC:=" , False,
      "ResimulateDC:=" , False,
      "MaxSolutions:=" , 250,
      "InterpUseSMatrix:=" , True,
      "InterpUsePortImpedance:=", True,
      "InterpUsePropConst:=" , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "InterpUseFullBasis:=" , True,
    "AdvDCExtrapolation:=" , False,
    "MinSolvedFreq:=" , "0.01GHz",
    "AutoSMatOnlySolve:=" , True,
    "MinFreqSMatrixOnlySolve:=", "1MHz",
    "CustomFrequencyString:=", "",
    "AllEntries:=" , False,
    "AllDiagEntries:=" , False,
    "AllOffDiagEntries:=" , False,
    "MagMinThreshold:=" , 0.01,
    "InterpMinSubranges:=" , 1,
    "InterpMinSolutions:=" , 0,
    "FreqSweepType:=" , "kDiscrete"
  ])
```

Analyze

Solves a single solution setup and all of its frequency sweeps.

UI Access	Right-click a solution setup in the project tree, and select **Analyze**.
Parameters	

Name	Type	Description
<setup>	String	Setup name.
isBlocking	Boolean	An optional arg that defaults to `true`. If it's `false`, the command immediately

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	Integer value: • **0** – Success • **Else** – Error		

Python Syntax	Analyze (<*setup*>)
Python Example	`oDesign.Analyze("Setup1", isBlocking)`

Analyze (Sweep)

Simulates a sweep.

UI Access	Right-click a sweep in the project tree, and select **Analyze**.		
Parameters	Name	Type	Description
	<*setup_sweep*>	String	Name of the selected setup and sweep, separated by a colon (e.g., Setup1 : Sweep1).
Return Value	None		

Python Syntax	Analyze(<setup_sweep>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDesign.Analyze("Setup1 : Sweep1")`

Delete (Layout Editor)

Deletes a setup.

UI Access	From the **Project Manager** window, right-click a setup and select **Delete**.
Parameters	Name | Type | Description <SetupName> | String | Name of the setup.
Return Value	None

Python Syntax	oModule.Delete(<SetupName>)
Python Example	`oModule.Delete("Q3D2")`

DeleteGFD (Setup HFSS 3D Layout)

Deletes the specified GFD (i.e., Green's Function Data).

UI Access	None.
Parameters	Name | Type | Description <*GFDName*> | String | Name of Green's Function Data file in the active design.
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DeleteGFD(<GFDName>)
Python Example	`oEditor.DeleteGFD("GDS_File.gds")`

DeleteSweep (Layout Editor)

Deletes a sweep.

UI Access	From the **Project Manager** window, right-click a setup and select **Delete**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><SweepName></td><td>String</td><td>Name of the sweep.</td></tr></table>
Return Value	None

Python Syntax	oModule.DeleteSweep(<SetupName>, <SweepName>)
Python Example	`oModule.DeleteSweep("Q3D2", "Sweep2")`

DynamicMeshOverlays (Layout Editor)

Use: Turns on dynamic mesh overlays

Syntax: DynamicMeshOverlays(STRING setup) // setup name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit (Layout Editor)

Edits a setup.

> **Note:** The following tables display **example** parameters applicable to a Near Field solution setup. Selecting a different setup will also add, remove, or change the applicable parameters.

<table>
<tr><td>UI Access</td><td>From the Project Manager window, right-click a setup and select Properties.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SetupArray></td><td>Array</td><td>An array that contains these parameters in order:
<NAME:SetupName>
<NameArray>
<CustomSetup>
<Position>
<SimSetupType>
<SimSettingsArray>
<SolveSetupType>
This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.</td></tr>
<tr><td><NAME:SetupName></td><td>String</td><td>User-entered name of the setup.</td></tr>
<tr><td><NameArray></td><td>Array</td><td>An array that contains these parameters in order:
<NAME:Properties>
<Enable></td></tr>
<tr><td><NAME:Properties></td><td>String</td><td>Identifies the contents of the array (i.e., Properties).</td></tr>
<tr><td><Enable></td><td>Boolean</td><td>True – setup is active
False – setup is inactive</td></tr>
<tr><td><SimSetupType></td><td>String</td><td>Identifies the type of setup (e.g., kSIwaveNearField).</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<SimSettingArray>	Array	*Optional*. An array that contains these parameters in order: • <NAME:SimulationSettings> • <Enabled> • <UseSISettings> • <UseCustomSettings> • <SISliderPros> • <PISliderPros> • <SIWAdvancedArray> • <SIWDCArray> • <SIWDCAdvancedArray> • <SIWSParamArray> • <NearFieldSettingArray> • <NAME:SweepDataList> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<NAME:SimulationSettings>	String	Identifies the contents of the array (i.e., SimulationSettings).
	<Enabled>	Boolean	**True** – array settings are active **False** – array settings are inactive
	<UseSISettings>	Boolean	**True** – uses SI Simulation settings **False** – uses PI Simulation settings
	<UseCustomSettings>	Boolean	**True** – overrides <UseSISettings> and activates <SIWAdvancedArray> parameters **False** groups by range
	<SISliderPros>	Integer	**0** – optimizes simulation for speed **1** – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
	<PISliderPros>	Integer	**0** – optimizes simulation for speed

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		1 – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
<SIWAdvancedArray>	Array	*Optional*. An array that contains these parameters in order: • <NAME:SIWAdvancedSettings> • <IncludeCoPlaneCoupling> • <IncludeInterPlaneCoupling> • <IncludeSplitPlaneCoupling> • <IncludeFringeCoupling> • <IncludeTraceCoupling> • <XtalkThreshold> • <MaxCoupledLines> • <MinVoidArea> • <MinPadAreaToMesh> • SnapLengthThreshold> • <MeshAutoMatic> • <MeshFrequency> • <ReturnCurrentDistribution> • <INcludeVISources> • <IncludeInfGnd> • <InfGndLocation> • <PerformERC> • <IgnoreNonFunctionalPads> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SIWAdvancedSettings>	String	Identifies the contents of the array (i.e., SIWAdvancedSettings).
<IncludeCoPlaneCoupling>	Boolean	**True** – coplane coupling active **False** – coplane coupling inactive

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<IncludeInterPlaneCoupling>	Boolean	**True** – intra-plane coupling active **False** – intra-plane coupling inactive
<IncludeSplitPlaneCoupling>	Boolean	**True** – split-plane coupling active **False** – split-plane coupling inactive
<IncludeFringeCoupling>	Boolean	**True** – cavity field coupling active **False** – cavity field coupling inactive
<IncludeTraceCoupling>	Boolean	**True** – trace coupling active **False** – trace coupling inactive
<XtalkThreshold>	Integer	Maximum cross-talk threshold, in decibels.
<MaxCoupledLines>	Integer	The number of allowed coupled lines.
<MinVoidArea>	Integer	Minimum size of voids; voids larger than the minimum are meshed.
<MinPadAreaToMesh>	Integer	Minimum size of pads; pads larger than the minimum are meshed.
SnapLengthThreshold>	Integer	Minimum distance of vertices; vertices less than the minimum are snapped.
<MeshAutoMatic>	Boolean	**True** – mesh are automatically refined **False** – mesh are not automatically refined
<MeshFrequency>	Integer	Specific frequency at which mesh are refined.
<ReturnCurrentDistribution>	Boolean	**True** – returns current distribution is traced **False** – returns current distribution is not traced
<IncludeVISources>	Boolean	**True** – parasitic elements from voltage/current sources are included **False** – parasitic elements are not included
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
<InfGndLocation>	Integer	Distance below bottom layer in stackup to place infinite ground plane.
<PerformERC>	Boolean	**True** – performs ERC during simulation **False** – does not perform ERC during simulation

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<IgnoreNonFunctionalPads>	Boolean	**True** – excludes non-functional pads from resonant mode, frequency sweep, and SYZ simulations **False** – does not exclude non-functional pads from resonant mode, frequency sweep, and SYZ simulations.
<SIWSParamSettings>	Array	*Optional*. An array that contains these parameters in order: • <UseStateSpace> • <Interpolation> • <Extrapolation> • <DCBehavior> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<UseStateSpace>	Boolean	**True** – uses the **State Space Model** **False** – uses a **Custom Model**
<Interpolation>	Integer	**0** – sets **Interpolation** to **Point** **1** – sets **Interpolation** to **Point** **2** – sets **Interpolation** to **Point**
<Extrapolation>	Integer	**0** – sets **Extrapolation** to **Zero padding** **1** – sets **Extrapolation** to **Same as last point** **2** – sets **Extrapolation** to **Linear extrapolation from last 2 points** **3** – sets **Extrapolation** to **Constant magnitude linear phase extrapolation**
<DCBehavior>	Integer	**0** – sets **DC Behavior** to **Zero padding** **1** – sets **DC Behavior** to **Same as last point** **2** – sets **DC Behavior** to **Linear extrapolation from last 2 points** **3** – sets **DC Behavior** to **Constant magnitude linear phase extrapolation** **4** – sets **DC Behavior** to **Series Capacitor** **5** – sets **DC Behavior** to **Leave all signal lines open circuited**
<SIWSNearFieldArray>	Array	An array that contains these parameters in order:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <NAME:SIWNearFieldSettings> • <UseExternalSources> • <ExternalSourceFilePath> • <InterpolateSpectrum> • <IncludeInfGnd> • <PosXoffset> • <NegXoffset> • <PosYoffset> • <NegYoffset> • <PosZoffset> • <NegZoffset> • <MinAdaptPasses> • <MaxAdaptPasses> • <NewTrianglesPerPass> • <GlobalErrorTol> • <UseUserEdgeLength> • <MeshEdgeLength> • <ExportData> • <ExportDataPath> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<NAME:SIWNearFieldSettings>	String	Identifies the contents of the array (i.e., SIWNearFieldSettings).
	<UseExternalSources>	Boolean	**True** – uses external **Excitations** definitions **False** – uses **Excitations** definitions defined in project
	<ExternalSourceFilePath>	String	Directory path to external **Excitations** definitions file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<InterpolateSpectrum>	Boolean	**True** – interpolates spectrum at missing frequency points (external **Excitations** definitions file only) **False** – does not interpolate spectrum at missing frequency points
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
<PosXoffset>	Integer	Value determines the +X offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegXoffset>	Integer	Value determines the -X offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<PosYoffset>	Integer	Value determines the +Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegYoffset>	Integer	Value determines the -Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<PosZoffset>	Integer	Value determines the +Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegZoffset>	Integer	Value determines the -Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<MinAdaptPasses>	Integer	Minimum number of passes the adaptive analysis will make, even if convergence criteria are met.
<MaxAdaptPasses>	Integer	Maximum number of passes the adaptive analysis will make, unless convergence criteria are met.
<NewTrianglesPerPass>	Integer	Percentage of triangles to add per pass.
<GlobalErrorTol>	Integer	Maximum relative number of errors to allow between two successive adaptive passes.
<UseUserEdgeLength>	Boolean	**True** – uses custom maximum edge length **False** – automatically determines maximum edge length
<MeshEdgeLength>	Integer	Maximum edge length.
<ExportData>	Boolean	**True** – exports Near Field data **False** – does not export Near Field data
<ExportDataPath>	String	Path to directory in which to save Near Field data.
<NAME:SweepDataList>	String	Identifies the determining factors of the Near Field simulation

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			(i.e., sweeps) (i.e., SweepDataList).
	<SolveSetupType>	String	Identifies the type of setup (i.e., SIwaveNearField).
Return Value	None		

Python Syntax	oModule.Edit([<NAME:SIwaveNearField1>, [<NAME:Properties>, <Enable>], <SimSetupType>, [<NAME:SimulationSettings>, <Enabled>, <UseSISettings>, <UseCustomSettings>, <SISliderPos>, <PISliderPos>], [<NAME:SIWAdvancedSettings>, <IncludeCoPlaneCoupling>, <IncludeInterPlaneCoupling>, <IncludeSplitPlaneCoupling>, <IncludeFringeCoupling>, <IncludeTraceCoupling>, <XtalkThreshold>, <MaxCoupledLines>, <MinVoidArea>, <MinPadAreaToMesh>, <SnapLengthThreshold>, <MeshAutoMatic>, <MeshFrequency>, <ReturnCurrentDistribution>, <IncludeVISources>, <IncludeInfGnd>, <InfGndLocation>, <PerformERC>, <IgnoreNonFunctionalPads>], [<NAME:SIWSParamSettings>, <UseStateSpace>, <Interpolation>, <Extrapolation>, <DCBehavior>], [<NAME:SIWNearFieldSettings>, <UseExternalSources>, <ExternalSourceFilePath>, <InterpolateSpectrum>, <IncludeInfGnd>, <PosXoffset>, <NegXoffset>, <PosYoffset>, <NegYoffset>, <PosZoffset>, <NegZoffset>, <MinAdaptPasses>, <MaxAdaptPasses>, <NewTrianglesPerPass>, <GlobalErrorTol>, <UseUserEdgeLength>, <MeshEdgeLength>, <ExportData>, <ExportDataPath>]], [<NAME:SweepDataList>], <SolveSetupType>])

Python Example	`oModule.Edit(`
	`  [`
	`  "NAME:SIwaveNearField1",`
	`  [`
	`  "NAME:Properties",`
	`  "Enable:="  , "true"`
	`  ],`
	`  "SimSetupType:="  , "kSIwaveNearField",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:SimulationSettings",
          "Enabled:="             , True,
          "UseSISettings:="       , False,
          "UseCustomSettings:="   , False,
          "SISliderPos:="         , 2,
          "PISliderPos:="         , 0,
        [
          "NAME:SIWAdvancedSettings",
          "IncludeCoPlaneCoupling:=", False,
          "IncludeInterPlaneCoupling:=", False,
          "IncludeSplitPlaneCoupling:=", False,
          "IncludeFringeCoupling:=", False,
          "IncludeTraceCoupling:=", False,
          "XtalkThreshold:="      , "-34",
          "MaxCoupledLines:="     , 12,
          "MinVoidArea:="         , "2mm2",
          "MinPadAreaToMesh:="    , "1mm2",
          "MinPlaneAreaToMesh:="  , "6.25e-6mm2",
          "SnapLengthThreshold:=" , "2.5um",
          "MeshAutoMatic:="       , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "MeshFrequency:="       , "4GHz",
        "ReturnCurrentDistribution:=", False,
        "IncludeVISources:="    , False,
        "IncludeInfGnd:="       , False,
        "InfGndLocation:="      , "0mm",
        "PerformERC:="          , False,
        "IgnoreNonFunctionalPads:=", True
      ],
      [
        "NAME:SIWSParamSettings",
        "UseStateSpace:="       , True,
        "Interpolation:="       , 1,
        "Extrapolation:="       , 3,
        "DCBehavior:="          , 3
      ],
      [
        "NAME:SIWNearFieldSettings",
        "UseExternalSources:="  , False,
        "ExternalSourceFilePath:=", "",
        "InterpolateSpectrum:=" , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "IncludeInfGnd:="       , False,
        "PosXoffset:="          , "1mm",
        "NegXoffset:="          , "1mm",
        "PosYoffset:="          , "1mm",
        "NegYoffset:="          , "1mm",
        "PosZoffset:="          , "1mm",
        "NegZoffset:="          , "1mm",
        "MinAdaptPasses:="      , 1,
        "MaxAdaptPasses:="      , 10,
        "NewTrianglesPerPass:=" , 20,
        "GlobalErrorTol:="      , 3,
        "UseUserEdgeLength:="   , False,
        "MeshEdgeLength:="      , "1mm",
        "ExportData:="          , False,
        "ExportDataPath:="      , ""
      ]
      ],
      [
        "NAME:SweepDataList"
      ],
      "SolveSetupType:="      , "SIwaveNearField"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

EditSweep (Layout Editor)

Edits a sweep.

UI Access	From the **Project Manager** window, right-click a setup and select **Add Frequency Sweep**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><Parameters></td><td>Array</td><td>A structured array containing the name and parameters for the frequency sweep.</td></tr></table>
Return Value	None

Python Syntax	oModule.EditSweep(<SetupName>, [<Parameters>])

Python Example	`oModule.EditSweep("Q3D1",`
	`  [`
	`  "NAME:Sweep1",`
	`  [`
	`  "NAME:Properties",`
	`  "Enable:=" , "true"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
      "NAME:Sweeps",
      "Variable:=" , "Sweep1",
      "Data:=" , "LIN 10MHz 100MHz 10MHz",
      "OffsetF1:=" , False,
      "Synchronize:=" , 0
    ],
    "GenerateSurfaceCurrent:=", False,
    "SaveRadFieldsOnly:=" , False,
    "ZoSelected:=" , False,
    "SAbsError:=" , 0.005,
    "ZoPercentError:=" , 1,
    "GenerateStateSpace:=" , False,
    "EnforcePassivity:=" , False,
    "PassivityTolerance:=" , 0.0001,
    "UseQ3DForDC:=" , False,
    "ResimulateDC:=" , False,
    "MaxSolutions:=" , 250,
    "InterpUseSMatrix:=" , True,
    "InterpUsePortImpedance:=", True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "InterpUsePropConst:=" , True,
    "InterpUseFullBasis:=" , True,
    "AdvDCExtrapolation:=" , False,
    "MinSolvedFreq:=" , "0.01GHz",
    "AutoSMatOnlySolve:=" , True,
    "MinFreqSMatrixOnlySolve:=", "1MHz",
    "CustomFrequencyString:=", "",
    "AllEntries:=" , False,
    "AllDiagEntries:=" , False,
    "AllOffDiagEntries:=" , False,
    "MagMinThreshold:=" , 0.01,
    "InterpMinSubranges:=" , 1,
    "InterpMinSolutions:=" , 0,
    "FreqSweepType:=" , "kDiscrete"
  ])
```

ExportToHFSS (Layout Editor)

Use: Exports an HFSS 3D Layout design to a new project with an equivalent HFSS 3D design. This command is only valid on HFSS Setup types.

Syntax: ExportToHfss(<setupName>, <pathToExportFile>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportToQ3D (Layout Editor)

Use: Exports an HFSS 3D Layout design to a new project containing an equivalent Q3D design. This command is only valid on HFSS Setup types.

Syntax: ExportToQ3d(<setupName>, <pathToExportFile>)

GetAllSolutionNames (Layout Editor)

Use: Returns a list of Ensemble solution names

Syntax: GetAllSolutionNames

Return Value: Array of Strings

Parameters: None

GetGFDData (Setup HFSS 3D Layout)

Shows all available GFD (i.e., Green's Function Data) files.

UI Access	None.
Parameters	None.
Return Value	Array of strings containing the names of identified Green's Function Data files.

Python Syntax	GetGFDData()
Python Example	`oEditor.GetGDData()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetMeshOperations

Get the names of all mesh operations for a given setup.

UI Access	N/A
Parameters	None.
Return Value	Array of names of all mesh operations.

Python Syntax	GetMeshOperations

Python Example	`oModule.GetMeshOperations()`
	`res = oModule.GetMeshOperations("Setup1")`
	`['Length 1', 'Length 2', 'SkinDepth 1']`

GetSetupData (Layout Editor)

Get setup data.

Command: None

Syntax: GetSetupData <setup>

Return Value: String array of setup information.

Parameters: <setup>

Type: string

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

LayoutMeshOverlay (Layout Editor)

Use: Overlay a mesh for a setup in layout

Syntax: LayoutMeshOverlay(STRING setup) // setup name

Layout3DMeshOverlay (Layout Editor)

Use: Overlay a mesh for a setup in 3D

Syntax: Layout3DMeshOverlay(STRING setup) // setup name

Example: `oModule. Layout3DMeshOverlay <setup>`

ListVariations (Layout Editor)

Use: Get a list of solution variations for a given solution

Syntax: ListVariations(STRING SolutionName) // solution name

Example: `oModule.ListVariations("NWA1")`

PullExcitationsFromTransient (Layout Editor)

Uses a Circuit or HFSS 3D Layout transient excitation source as an external source for a near field simulation. Includes optional transient parameters to control harmonic extraction and excitation filtering.

UI Access	None		
Parameters	Name	Type	Description
	<SetupName>	String	Name of the target setup (e.g., SIwaveDCIR1) where excitations will be applied.
	<ExcitationSource>	String	Name of the transient simulation setup (e.g., NexximTransient1) used as the excitation source.
	<EnableExternalExcitation>	Boolean	**True** – use transient excitation as external source. **False** – do not use transient excitation.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<NAME: TransientParameters>	Array	Optional parameter block containing the following transient extraction settings: <StartTime> <StopTime> <MaxHarmonicOrder> <RelativeCurrThresh> <MinCurrMag> <WinType> <KaiserParam>
<StartTime>	Integer	Start time (in seconds) of the transient data window used for frequency-domain extraction.
<<StopTime>	Integer	End time (in seconds) of the transient data window used for frequency-domain extraction.
<MaxHarmonicOrder>	Integer	Maximum number of harmonics to extract from the transient waveform for frequency-domain conversion.
<RelativeCurrThresh>	Integer	Minimum relative current threshold (as a fraction of peak current) for including a source in the extracted data.
<MinCurrMag>	Integer	Minimum absolute current magnitude (in Amps) required for a source to be included in the frequency-domain data.
<WinType>	Integer	Type of windowing function applied to the transient waveform before FFT. Integer values correspond to: • **0** – Rectangular • **1** – Bartlett • **2** – Blackman • **3** – Hamming • **4** – Hanning

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• **5** – Kaiser • **6** – Welch • **7** – Weber • **8** – Lanczos
	<KaiserParam>	Integer	Shape parameter for the Kaiser window. Controls tradeoff between main lobe width and side lobe suppression. Only used if *WinType* is set to Kaiser.
Return Value	None		

Python Syntax	oModule.PullExcitationsFromTransient(<SetupName>, <ExcitationSource>, <EnableExternalExcitation>, <TransientParameters>)

Python Example	`oModule.PullExcitationsFromTransient("SIwaveDCIR1", "NexximTransient1", False, [`
	`  "NAME:transient parameters",`
	`  "StartTime:=", 0,`
	`  "StopTime:=", 2E-08,`
	`  "MaxHarmonicOrder:=", 100,`
	`  "RelativeCurrThresh:=", 0.01,`
	`  "MinCurrMag:=", 1E-05,`
	`  "WinType:=", 3,`
	`  "KaiserParam:=", 0`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RefreshMeshOverlays (Layout Editor)

Use: Refreshes mesh display for a setup

Syntax: RefreshMeshOverlays(STRING setup) // setup name

Example: `oModule.RefreshMeshOverlays <setup>`

Rename (Layout Editor)

Renames a setup.

UI Access	From the **Project Manager** window, right-click a setup and select **Rename**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Current name of the setup.</td></tr><tr><td><NewName></td><td>String</td><td>User-entered new name of the setup.</td></tr></table>
Return Value	None

Python Syntax	oModule.Rename(<SetupName>, <NewName>)
Python Example	`oModule.Rename("Q3D1", "Q3D2")`

RenameSweep (Layout Editor)

Renames a simulation sweep.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	From the **Project Manager** window, expand a setup. Then right-click a sweep and select **Rename**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><SweepName></td><td>String</td><td>Current name of the sweep.</td></tr><tr><td><NewSweepName></td><td>String</td><td>User-entered new name of the sweep.</td></tr></table>
Return Value	None

Python Syntax	oModule.RenameSweep(<SetupName>, <SweepName>, <NewSweepName>)
Python Example	`oModule.RenameSweep("Q3D2", "Sweep1", "Sweep2")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

22 - Excitations Commands

The following commands are available in the Excitations module. These commands should be executed by the **oDesign** object. For example:

```
oDesign.GetModule("Excitations")

oModule.CommandName <args>
```

The following commands are described in this section:

Add

AddRefPort

AddRefPortUsingEdges

AlphaNumericMatrix

ConvertPlaneToTrace

CoupleEdgePorts

DecoupleEdgePorts

Delete

DeleteExcitation

DeleteSource

DeleteProbePortAndVia

Edit

EditCircuitPort

EditExcitations

EditHorizontalWavePort

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetAllBoundariesList

GetAllPortsList

RemoveRefPort

Rename

RenameSource

ReorderMatrix

SelectInLayout

Add (Excitations)

Adds an excitation to the design.

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing <excitation_data>.</td></tr><tr><td><*excitation_data*></td><td>String</td><td>Required excitation data.</td></tr></table>
Return Value	None

Python Syntax	Add([<excitation_data>])
Python Example	`oModule.Add(["excitation_data"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddRefPort (Layout Editor)

Creates a reference port from edges and associates the reference port with each of the named ports.

<table>
<tr><td>UI Access</td><td>Right-click within the Layout Editor and select Port > Create.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Port_List></td><td>Array</td><td>Structured array containing one or more Port_Name strings. Each Port_Name represents a named port (e.g., "port1", "port2", "port3").</td></tr>
<tr><td><Port_Name></td><td>String</td><td>Individual port name string used within the Port_List array.</td></tr>
<tr><td><Edge_Array></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Contents> — Required string with value "Contents"
<Edge> — One entry per Port_Name in Port_List
Each Edge entry may reference a predefined edge object (e.g., "edge_description") or contain a full Edge_Description array inline.</td></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Contents").</td></tr>
<tr><td><Edge></td><td>Array</td><td>Structured array containing an <Edge_description> parameter for each <Port_Name> parameter provided in the <Port List> array.</td></tr>
<tr><td><Edge_ Description></td><td>Array</td><td>Name of the array containing the geometry of the selected edge, containing these parameters in order:
<Layer>
<Port>
<Et>
<Pe></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <Prim> • <Edge> • <Edge#> Via edges: • <Et> • <Pse> • <Sel> • <Via> • <Layer> • <Layer_ID> • <Sx> • <Sy> • <Ex> • <Ey> • <H> • <Rad>
<Port>	String	Name of the selected port or pin.
<Et>	String	Specifies the edge type: Pe for primitive edge or Pse for via edge.
<Prim>	String	Specifies the primitive type associated with the edge.
<Edge#>	Integer	Index number of the selected primitive.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Sel>	Integer	Selection index for the via. Used to identify which via instance is associated with the edge. Typically set to 0 unless multiple vias are present.
	<Via>	String	Name of the selected via.
	<Layer>	Integer	The ID of the layer of the pad of the via to use.
	<Sx>	Integer	X coordinate of the start point of the edge arc.
	<Sy>	Integer	Y coordinate of the start point of the edge arc.
	<Ex>	Integer	X coordinate of the end point of the edge arc.
	<Ey>	Integer	Y coordinate of the end point of the edge arc.
	<H>	Integer	Height of the edge arc (0 for a straight edge).
	<Rad>	Integer	Size of the edge arc in radians (0 for a straight edge).
Return Value	None		

Python Syntax	AddRefPort([<port_list>], [<edge_array>])

Python Example	`oEditor.AddRefPort([`
	`  "port1",`
	`  "port2"],`
	`  ["NAME:Contents",`
	`  "edge:=" , "edge_description",`
	`  "edge:=" , "edge_description"]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddRefPortUsingEdges (Layout Editor)

Use: Creates an edge port with reference using the specified edge

Command: None

Syntax: <port name>,

Array("Name:EdgeRefs",

"edge:=", Array(<edge information>)

Return Value: None

Parameters: <port name>: string that will name the created edge port

<edge information>: describes the edge.

There are two choices: primitive edge or via edge.

<edge information> for a primitive edge:

"et:=", "pe", "prim:=", <primitive name>, "edge:=", <edge number>

<primitive name>: text that is the name of the primitive to use

<edge number>: an integer indexing the edge of the primitive to use

<edge information> for via edge:

"et:=", "pse", "sel:=", <via name>, "layer:=", <layer id>,

"sx:=", <start X location>, "sy:=", <start Y location>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"ex:=", <end X location>, "ey:=", <end Y location>,

"h:=", <arc height>, "rad:=", <radians>

<via name>: text that is the name of the via to use

<layer id>: an integer that is the id of the layer of the pad of the via to use

<start X location>, <start Y Location>:

doubles that are the X, Y location of the start point of the edge arc

<end X location>, <end Y Location>:

doubles that are the X, Y location of the end point of the edge arc

<arc height>: double giving the height of the edge arc (0 for a straight edge)

<radians>: double giving the arc size in radians (0 for a straight edge)

```
oModule.AddRefPortUsingEdges "Port1", Array("NAME:EdgeRefs", "edge:=",
Array("et:=", "pe", "prim:=", "rect_3", "edge:=", 1))

oModule.AddRefPortUsingEdges "Port1", Array("NAME:EdgeRefs", "edge:=",
Array("et:=", "pse", "sel:=", "via_5", "layer:=", 10, "sx:=", 0.0015, "sy:=", 0.0015,
"ex:=", -0.0015, "ey:=", 0.0015, "h:=", 0, "rad:=", 0))
```

ConvertPlaneToTrace (Layout Editor)

Converts selected polygon planes into trace objects.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click a polygon, then select **Draw** > **Planes to Traces**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Polygon_Data*></td><td>Array</td><td>List of polygon objects to convert. Each entry should reference a valid polygon name.</td></tr></table>
Return Value	None

Python Syntax	ConvertPlaneToTrace(<*Polygon_Data*>)

Python Example	`oEditor.ConvertPlaneToTrace(`
	`  ["polygon_01",`
	`  "polygon_02"`
	`  ])`

ConvertTraceToPlane (Layout Editor)

Converts selected trace paths into polygon plane objects.

UI Access	Right-click a path, then select **Draw** > **Traces to Planes**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Line_Data*></td><td>Array</td><td>List of trace path objects to convert. Each entry should reference a valid line or path name.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ConvertTraceToPlane(<*Line_Data*>)

Python Example	`oEditor.ConvertTraceToPlane(`
	`  ["trace_01",`
	`  "trace_02"`
	`  ])`

CoupleEdgePorts

Use: Couples valid edge ports

Syntax: CoupleEdgePorts(VARIANT coupledlist) //entry containing ports to be coupled

sure

CreateGroundPortComponent (Layout Editor)

Creates a ground port component at unconnected pins of the selected port.

UI Access	From the **Layout Editor**, right-click a port and select **Port** > **Nexxim Ports** > **Add Grounds at Unconnected Pins**.		
Parameters	Name	Type	Description
	<*Component_Data*>	Array	A structured array consisting of ground port data.
Return Value	None		

Python Syntax	oEditor.CreateGroundPortComponent ([<Component_Data>])
Python Example	`oEditor.CreateGroundPortComponent(Component_Data)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateInterfaceGround (Layout Editor)

Use: Create an interrface ground

Command: Right-click selected port > Port > Nexxim Ports > Circuit Ground

Return Value: None

Parameters: `<Port_Data>`

Type: array

CreateInterfacePort (Layout Editor)

Use: Create an interrface port

Command: Right-click selected port > Port > Nexxim Ports > Circuit Interface

Syntax: CreateInterfacePort <Port_Data>

Parameters: `<Port_Data>`

Type: array

CreateInterfacePortComponent (Layout Editor)

Use: Create an interface port component

Command: Right-click selected port > Port > Nexxim Ports > Add Interface Ports at Unconnected Pins

Syntax: CreateInterfacePortComponent <Port_Data>

Parameters: `<Component_Data>`

Type: array

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DecoupleEdgePorts

Use: Decouples edge ports

Syntax: DecoupleEdgePorts(

VARIANT decoupledlist) //entry containing ports to be decoupled

Delete[Excitation HFSS 3D Layout]

Use: Deletes an excitation

Syntax: Delete(STRING excitation_name) // excitation name

Example: oModule. Delete <excitation_name>

Delete [Interface Port]

Use: Deletes an interface port

Syntax: Delete(STRING: Interface Port name)

Delete[Excitation HFSS 3D Layout]

Use: Deletes an excitation

Syntax: Delete(STRING excitation_name) // excitation name

Example: oModule. Delete <excitation_name>

DeleteProbePortAndVia (Layout Editor)

Use: Deletes a coaxial probe port and the associated via

Syntax: DeleteProbePortAndVia(STRING port_name) // Probe Port name

STRING via_name) //Via name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: oModule. DeleteProbePortAndVia <port_name> <via_name>

DeleteSource [Interface Source]

Use: Deletes an interface source

Syntax: DeleteSource (STRING: Interface Source name)

Edit (Layout Editor)

Edits a setup.

> **Note:** The following tables display **example** parameters applicable to a Near Field solution setup. Selecting a different setup will also add, remove, or change the applicable parameters.

<table>
<tr><td>UI Access</td><td>From the Project Manager window, right-click a setup and select Properties.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SetupArray></td><td>Array</td><td>An array that contains these parameters in order:
<NAME:SetupName>
<NameArray>
<CustomSetup>
<Position>
<SimSetupType>
<SimSettingsArray>
<SolveSetupType></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SetupName>	String	User-entered name of the setup.
<NameArray>	Array	An array that contains these parameters in order: • <NAME:Properties> • <Enable>
<NAME:Properties>	String	Identifies the contents of the array (i.e., Properties).
<Enable>	Boolean	**True** – setup is active **False** – setup is inactive
<SimSetupType>	String	Identifies the type of setup (e.g., kSIwaveNearField).
<SimSettingArray>	Array	*Optional*. An array that contains these parameters in order: • <NAME:SimulationSettings> • <Enabled> • <UseSISettings> • <UseCustomSettings> • <SISliderPros> • <PISliderPros> • <SIWAdvancedArray> • <SIWDCArray> • <SIWDCAdvancedArray> • <SIWSParamArray> • <NearFieldSettingArray> • <NAME:SweepDataList> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SimulationSettings>	String	Identifies the contents of the array (i.e., SimulationSettings).
<Enabled>	Boolean	**True** – array settings are active **False** – array settings are inactive

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<UseSISettings>	Boolean	**True** – uses SI Simulation settings **False** – uses PI Simulation settings
<UseCustomSettings>	Boolean	**True** – overrides <UseSISettings> and activates <SIWAdvancedArray> parameters **False** groups by range
<SISliderPros>	Integer	**0** – optimizes simulation for speed **1** – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
<PISliderPros>	Integer	**0** – optimizes simulation for speed **1** – optimizes simulation to balance speed and accuracy **2** – optimizes simulation for accuracy
<SIWAdvancedArray>	Array	*Optional*. An array that contains these parameters in order: • <NAME:SIWAdvancedSettings> • <IncludeCoPlaneCoupling> • <IncludeInterPlaneCoupling> • <IncludeSplitPlaneCoupling> • <IncludeFringeCoupling> • <IncludeTraceCoupling> • <XtalkThreshold> • <MaxCoupledLines> • <MinVoidArea> • <MinPadAreaToMesh> • SnapLengthThreshold> • <MeshAutoMatic> • <MeshFrequency> • <ReturnCurrentDistribution> • <INcludeVISources>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <IncludeInfGnd> • <InfGndLocation> • <PerformERC> • <IgnoreNonFunctionalPads> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SIWAdvancedSettings>	String	Identifies the contents of the array (i.e., SIWAdvancedSettings).
<IncludeCoPlaneCoupling>	Boolean	**True** – coplane coupling active **False** – coplane coupling inactive
<IncludeInterPlaneCoupling>	Boolean	**True** – intra-plane coupling active **False** – intra-plane coupling inactive
<IncludeSplitPlaneCoupling>	Boolean	**True** – split-plane coupling active **False** – split-plane coupling inactive
<IncludeFringeCoupling>	Boolean	**True** – cavity field coupling active **False** – cavity field coupling inactive
<IncludeTraceCoupling>	Boolean	**True** – trace coupling active **False** – trace coupling inactive
<XtalkThreshold>	Integer	Maximum cross-talk threshold, in decibels.
<MaxCoupledLines>	Integer	The number of allowed coupled lines.
<MinVoidArea>	Integer	Minimum size of voids; voids larger than the minimum are meshed.
<MinPadAreaToMesh>	Integer	Minimum size of pads; pads larger than the minimum are meshed.
SnapLengthThreshold>	Integer	Minimum distance of vertices; vertices less than the minimum are snapped.
<MeshAutoMatic>	Boolean	**True** – mesh are automatically refined **False** – mesh are not automatically refined
<MeshFrequency>	Integer	Specific frequency at which mesh are refined.
<ReturnCurrentDistribution>	Boolean	**True** – returns current distribution is traced **False** – returns current distribution is not traced
<IncludeVISources>	Boolean	**True** – parasitic elements from voltage/current sources are

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		included **False** – parasitic elements are not included
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
<InfGndLocation>	Integer	Distance below bottom layer in stackup to place infinite ground plane.
<PerformERC>	Boolean	**True** – performs ERC during simulation **False** – does not perform ERC during simulation
<IgnoreNonFunctionalPads>	Boolean	**True** – excludes non-functional pads from resonant mode, frequency sweep, and SYZ simulations **False** – does not exclude non-functional pads from resonant mode, frequency sweep, and SYZ simulations.
<SIWSParamSettings>	Array	*Optional*. An array that contains these parameters in order: • <UseStateSpace> • <Interpolation> • <Extrapolation> • <DCBehavior> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<UseStateSpace>	Boolean	**True** – uses the **State Space Model** **False** – uses a **Custom Model**
<Interpolation>	Integer	**0** – sets **Interpolation** to **Point** **1** – sets **Interpolation** to **Point** **2** – sets **Interpolation** to **Point**
<Extrapolation>	Integer	**0** – sets **Extrapolation** to **Zero padding** **1** – sets **Extrapolation** to **Same as last point** **2** – sets **Extrapolation** to **Linear extrapolation from last 2 points** **3** – sets **Extrapolation** to **Constant magnitude linear phase extrapolation**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<DCBehavior>	Integer	**0** – sets **DC Behavior** to **Zero padding** **1** – sets **DC Behavior** to **Same as last point** **2** – sets **DC Behavior** to **Linear extrapolation from last 2 points** **3** – sets **DC Behavior** to **Constant magnitude linear phase extrapolation** **4** – sets **DC Behavior** to **Series Capacitor** **5** – sets **DC Behavior** to **Leave all signal lines open circuited**
	<SIWSNearFieldArray>	Array	An array that contains these parameters in order: • <NAME:SIWNearFieldSettings> • <UseExternalSources> • <ExternalSourceFilePath> • <InterpolateSpectrum> • <IncludeInfGnd> • <PosXoffset> • <NegXoffset> • <PosYoffset> • <NegYoffset> • <PosZoffset> • <NegZoffset> • <MinAdaptPasses> • <MaxAdaptPasses> • <NewTrianglesPerPass> • <GlobalErrorTol> • <UseUserEdgeLength> • <MeshEdgeLength> • <ExportData> • <ExportDataPath>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
<NAME:SIWNearFieldSettings>	String	Identifies the contents of the array (i.e., SIWNearFieldSettings).
<UseExternalSources>	Boolean	**True** – uses external **Excitations** definitions **False** – uses **Excitations** definitions defined in project
<ExternalSourceFilePath>	String	Directory path to external **Excitations** definitions file.
<InterpolateSpectrum>	Boolean	**True** – interpolates spectrum at missing frequency points (external **Excitations** definitions file only) **False** – does not interpolate spectrum at missing frequency points
<IncludeInfGnd>	Boolean	**True** – infinite ground plane is included **False** – infinite ground plane is not included
<PosXoffset>	Integer	Value determines the +X offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegXoffset>	Integer	Value determines the -X offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<PosYoffset>	Integer	Value determines the +Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegYoffset>	Integer	Value determines the -Y offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<PosZoffset>	Integer	Value determines the +Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<NegZoffset>	Integer	Value determines the -Z offset of a cube that encloses the design; the Near Field is evaluated on the surface of this cuboid.
<MinAdaptPasses>	Integer	Minimum number of passes the adaptive analysis will make, even if convergence criteria are met.
<MaxAdaptPasses>	Integer	Maximum number of passes the adaptive analysis will make, unless convergence criteria are met.
<NewTrianglesPerPass>	Integer	Percentage of triangles to add per pass.
<GlobalErrorTol>	Integer	Maximum relative number of errors to allow between two suc-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			cessive adaptive passes.
	<UseUserEdgeLength>	Boolean	**True** – uses custom maximum edge length **False** – automatically determines maximum edge length
	<MeshEdgeLength>	Integer	Maximum edge length.
	<ExportData>	Boolean	**True** – exports Near Field data **False** – does not export Near Field data
	<ExportDataPath>	String	Path to directory in which to save Near Field data.
	<NAME:SweepDataList>	String	Identifies the determining factors of the Near Field simulation (i.e., sweeps) (i.e., SweepDataList).
	<SolveSetupType>	String	Identifies the type of setup (i.e., SIwaveNearField).
Return Value	None		

Python Syntax	oModule.Edit([<NAME:SIwaveNearField1>, [<NAME:Properties>, <Enable>], <SimSetupType>, [<NAME:SimulationSettings>, <Enabled>, <UseSISettings>, <UseCustomSettings>, <SISliderPos>, <PISliderPos>], [<NAME:SIWAdvancedSettings>, <IncludeCoPlaneCoupling>, <IncludeInterPlaneCoupling>, <IncludeSplitPlaneCoupling>, <IncludeFringeCoupling>, <IncludeTraceCoupling>, <XtalkThreshold>, <MaxCoupledLines>, <MinVoidArea>, <MinPadAreaToMesh>, <SnapLengthThreshold>, <MeshAutoMatic>, <MeshFrequency>, <ReturnCurrentDistribution>, <IncludeVISources>, <IncludeInfGnd>, <InfGndLocation>, <PerformERC>, <IgnoreNonFunctionalPads>], [<NAME:SIWSParamSettings>, <UseStateSpace>, <Interpolation>, <Extrapolation>, <DCBehavior>], [<NAME:SIWNearFieldSettings>, <UseExternalSources>, <ExternalSourceFilePath>, <InterpolateSpectrum>, <IncludeInfGnd>, <PosXoffset>, <NegXoffset>, <PosYoffset>, <NegYoffset>, <PosZoffset>, <NegZoffset>, <MinAdaptPasses>, <MaxAdaptPasses>, <NewTrianglesPerPass>, <GlobalErrorTol>, <UseUserEdgeLength>, <MeshEdgeLength>, <ExportData>, <ExportDataPath>]], [<NAME:SweepDataList>], <SolveSetupType>])

Python Example	`oModule.Edit(`
	`  [`
	`  "NAME:SIwaveNearField1",`
	`  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:Properties",
        "Enable:="              , "true"
      ],
      "SimSetupType:="        , "kSIwaveNearField",
      [
        "NAME:SimulationSettings",
        "Enabled:="             , True,
        "UseSISettings:="       , False,
        "UseCustomSettings:="   , False,
        "SISliderPos:="         , 2,
        "PISliderPos:="         , 0,
      [
        "NAME:SIWAdvancedSettings",
        "IncludeCoPlaneCoupling:=", False,
        "IncludeInterPlaneCoupling:=", False,
        "IncludeSplitPlaneCoupling:=", False,
        "IncludeFringeCoupling:=", False,
        "IncludeTraceCoupling:=", False,
        "XtalkThreshold:="      , "-34",
        "MaxCoupledLines:="     , 12,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "MinVoidArea:="         , "2mm2",
        "MinPadAreaToMesh:="    , "1mm2",
        "MinPlaneAreaToMesh:="  , "6.25e-6mm2",
        "SnapLengthThreshold:=" , "2.5um",
        "MeshAutoMatic:="       , True,
        "MeshFrequency:="       , "4GHz",
        "ReturnCurrentDistribution:=", False,
        "IncludeVISources:="    , False,
        "IncludeInfGnd:="       , False,
        "InfGndLocation:="      , "0mm",
        "PerformERC:="          , False,
        "IgnoreNonFunctionalPads:=", True
      ],
      [
        "NAME:SIWSParamSettings",
        "UseStateSpace:="       , True,
        "Interpolation:="       , 1,
        "Extrapolation:="       , 3,
        "DCBehavior:="          , 3
      ],
      [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:SIWNearFieldSettings",
        "UseExternalSources:="  , False,
        "ExternalSourceFilePath:=", "",
        "InterpolateSpectrum:=" , True,
        "IncludeInfGnd:="       , False,
        "PosXoffset:="          , "1mm",
        "NegXoffset:="          , "1mm",
        "PosYoffset:="          , "1mm",
        "NegYoffset:="          , "1mm",
        "PosZoffset:="          , "1mm",
        "NegZoffset:="          , "1mm",
        "MinAdaptPasses:="      , 1,
        "MaxAdaptPasses:="      , 10,
        "NewTrianglesPerPass:=" , 20,
        "GlobalErrorTol:="      , 3,
        "UseUserEdgeLength:="   , False,
        "MeshEdgeLength:="      , "1mm",
        "ExportData:="          , False,
        "ExportDataPath:="      , ""
      ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	], ["NAME:SweepDataList"], "SolveSetupType:=" , "SIwaveNearField"])

EditCircuitExcitations (Layout Editor)

*Use:*Edits the properties of all the existing circuit excitations

*Syntax:*EditCircuitExcitations Array("NAME:NexximPostProcess", Array("NAME:portname", do-post-proc, "de-embedding distance", "re-normalization impedance")...))

*Return Value:*None

EditCircuitPort [Interface Port]

Use: Edit a circuit port

Command: None

Syntax: EditCircuitPort <Port Name> <Port Info> <Port Properties>

Return Value: None

Parameters: <Port Name> - Type: string

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<Port Info> - Type: array

<Port Properties> - Type:array

EditExcitations (Layout Editor)

Edits the properties of all existing excitations.

> **Note:** GapSource should always be set.

UI Access	None.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*exciteArray*></td><td>Structured Array</td><td>Excitations block.</td></tr><tr><td><*ppArray*></td><td>Structured Array</td><td>Post-processing block.</td></tr><tr><td><*gapSource*></td><td>String</td><td>Gap source.</td></tr><tr><td><pushExParams></td><td>Structured Array</td><td>PushExParams block. "" by default.</td></tr></table>
Return Value	None.

Python Syntax	EditExcitations (<*exciteArray*>, <*ppArray*>, <*gapSource*>, <pushExParams>))

Python Example	`oModule.EditExcitations(`
	`  [ "NAME:Excitations",`
	`  ["Port1","0deg","0.099V"],`
	`  ["Port2","0deg","0.099V"]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ],
  ["NAME:PostProcess",
    ["Port1",True,True,"0mm","50ohm + 0i ohm"],
    ["Port2",True,True,"0mm","50ohm + 0i ohm"]
  ],
  "GapSource,IncidentVoltage",
  [])
```

EditHorizontalWavePort (Layout Editor)

Edits an existing horizontal wave port by specifying the terminal name, alignment option, and the list of terminals currently enabled.

UI Access	Right-click within the **Layout Editor** and select **Port** > **Edit Horizontal Wave Port**.
Parameters	

Name	Type	Description
<*TerminalName*>	String	Name of the terminal the dialog was opened on (e.g., "Port1:1").
<*IsPortTopAligned*>	Boolean	Specifies whether the port is top aligned (`True` or `False`).
<*TerminalsInUse*>	Array	List of terminals currently enabled for the port (e.g., "via_0", "via_1", "via_2").

Return Value	None

Python Syntax	EditHorizontalWavePort(<TerminalName>, <IsPortTopAligned>, [<TerminalsInUse>])
Python Example	

```
oModule.EditHorizontalWavePort(
  "Port1:1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   False,
      [ "NAME:TerminalsInUse",
      "via_0",
      "via_1",
      "via_2" ]
)
```

GetAllBoundariesList (Layout Editor)

Use: Get a list of all boundaries.

Command: None

Syntax: GetAllBoundariesList <bounds>

Return Value: String array of list of all boundaries.

Parameters: <bounds>

Type: string

Example:

```
oEx = oDesign.GetModule('Excitations')
bounds = oEx.GetAllBoundariesList()
```

GetAllPortsList (Layout Editor)

Use: Returns the names of all Ensemble ports

Syntax: GetAllPortsList

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: Array of Strings

Parameters: None

Rename [Interface Port]

Use: Renames an interface port

Syntax: Rename (STRING: Interface Port name)(STRING: New name)

Example: `oModule.Rename "OldName" "NewName"`

RenameSource [Interface Source]

Use: Renames an interface source

Syntax: RenameSource (STRING: Interface Source name)

Example: `oModule.RenameSource "OldName" "NewName"`

Rename (Layout Editor)

Use: Renames an excitation

Syntax: Rename(STRING oldname) // old excitation name

STRING newname) // new excitation name

Example: `oModule. Rename <oldname> <newname>`

RemoveRefPort (Layout Editor)

Removes the reference for an existing port.

<table>
<tr><td>UI Access</td><td>Right click on port and select Port > Clear Reference</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><portName></td><td>String</td><td>Name of the port reference to be removed</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	RemoveRefPort(<portName>)
Python Example	`oModule.RemoveRefPort("W1")`

SelectInLayout (Layout Editor)

Use: Selects and highlights an excitation in the layout editor

Syntax: SelectInLayout(STRING excitation_name) // excitation name

Example: `oModule. SelectInLayout <excitation_name>`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

23 - 2.5D Via Commands

The following commands are available in the LayoutVias module. These commands should be executed by the **oDesign** object. For example:

```
oDesign.GetModule("LayoutVias")

oModule.CommandName <args>
```

The following commands are described in this section:

ConvertPrimitives

Delete

Edit

MultipleEdit

Rename

SelectInLayout

ConvertPrimitives (Layout Editor)

Converts the selected primitives into 2.5D via structures.

UI Access	None		
Parameters	Name	Type	Description
	<primvia_data>	Array	List of primitive objects to convert. Each entry should reference a valid primitive name.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ConvertPrimitives(<*primvia_data*>)

Python Example	`oEditor.ConvertPrimitives(`
	`  ["primitive_01",`
	`  "primitive_02"`
	`  ])`

Delete [Via HFSS 3D Layout]

Use: Delete a 2.5 via

Syntax: Delete(

STRING via) // 2.5 via name

Example: `oModule. Delete via`

Edit [Via HFSS 3D Layout]

Use: Edits a 2.5 via

Syntax: Edit(STRING via) // 2.5 via name

ARRAY via_data) // data for the 2.5 via

Example: `oModule. Edit via via_data`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

MultipleEdit [multiple via HFSS 3D Layout]

Use: Edit the properties of a collection of 2.5D via

Syntax: MultipleEdit(ARRAY multiplevia_data) // multiple 2.5 via data

Example: `oModule. MultipleEdit multiplevia_data`

Rename [Via HFSS 3D Layout]

Use: Renames a 2.5 via

Syntax: Rename(STRING oldname) // old 2.5 via name

`STRING newname) // new 2.5 via name`

Example: `oModule. Rename oldname newname`

SelectInLayout [Via HFSS 3D Layout]

Use: Selects and highlights the 2.5 via in the layout editor

Syntax: SelectInLayout(STRING via) // 2.5 via name

Example: `oModule. SelectInLayout via`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

24 - Layout Scripting

The Layout scripting interface is a set of commands that can be executed by the *Layout Editor* interface of *Design* objects. Accessing the *Layout Editor* interface can be done by the *SetActiveEditor* method of a *Design* object, for example:

```
Dim oEditor

oEditor = DesignObject.SetActiveEditor ("Layout")
```

The call to each scripting method contains three elements:

- Layout editor
- Method name
- Method arguments (if any)

For instance, in Visual Basic, a call might look like this:

```
oEditor.Delete ("rect_1")
```

The behavior of the scripting methods is similar to the corresponding commands in the Layout editor. This correspondence is particularly evident when recording a script in Layout Editor (**Tools>Record Script**).

> **Note:** A subsection of the Layout Scripting commands are only accessible once **Electronics Desktop** has been placed in IC Editor mode (i.e., **IC** mode). These commands are all located within the IC Editor Script Commands directory. There are commands common to both the **General** mode and **IC** mode environments in every topic within Layout Scripting, but they are designated **(General Mode and IC Mode)** at the **Table of Contents** level. Unless otherwise noted, common commands function identically in either environment (e.g., **RemoveLayer** is divided into two sections for each mode since the command function is similar but not identical).

The topics for this section include:

Object Identifiers and Script Recording

Create Primitives

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Create Voids in Primitives

Other Creation Methods

Object Movement and Modification Methods

Activation and Deactivation Methods

Layout and Geometry Interrogation Methods

Boolean Operations on Primitives

Coordinate System Methods

IC Editor Script Commands (IC Mode Only)

Layers Methods

NetClass Methods

Port Methods

Miscellaneous Methods

Object Identifiers and Script Recording

Many scripting methods apply to one or more *existing* objects. Objects in scripts are uniquely identified by their *names*, and the object names are typically referenced as:

["name1", "name2"] // name1 and name2 are the names of the objects.

While working interactively in the Layout Editor, the user may observe (but not modify, except in some special cases) the *names* of the currently selected objects through the "Selected Object Properties" dialogs. These same names are the ones that the scripting methods use in order to identify the object the method applies to.

When particular objects are involved in interactive script recording, they are typically selected before the actual execution the particular interactive command. In this case, the *selected objects*' names would appear within the generated script method argument(s).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Create Primitives

The following create primitives are available.

Create3DLine

CreateCircle

CreateLine

CreatePolygon

CreateRectangle

CreateText

AddObject ("3D Line") (Layout Editor)

Adds a 3D line to the active design.

<table>
<tr><td>UI Access</td><td>From the Layout ribbon, select Draw 3D line.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ObjectType></td><td>String</td><td>Type of object to add (i.e., must have the value "3D Line").</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Contents>
<3D LineGeometry></td></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Identifies the contents of the array (i.e., "Contents").</td></tr>
<tr><td><3D LineGeometry></td><td>Array</td><td>Structured array containing these parameters in order:
<Name>
<LayerName>
<lw>
<gm></td></tr>
<tr><td><Name></td><td>String</td><td>Name of the new object.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*LayerName*>	String	Name of the target layer.
	<*lw*>	Integer	Line width of the new object.
	<*gm*>	Array	Structured array containing these parameters in order: • <m> • <pnt>
	<*m*>	Integer	"Placement Mode" - species one of the following: **0** 3D - points are placed according to user-specified XYZ coordinates. **1** Top - points are placed on top of the target layer. **2** Middle - points are placed in the middle of the target layer. **3** Bottom - points are placed on bottom of the target layer.
	<*pnt*>	Array	Structured array containing these parameters in order: • <lid> • <pos>
	<*lid*>	Integer	Identifies the layer identification number on which the 3D line's position will be affixed (e.g., "2").
	<*pos*>	Array	Structured array containing these parameters in order: • <X> • <Y> • <Z>
	<*X*>	Integer	X coordinate of the new object (e.g., -0.08mm).
	<*Y*>	Integer	X coordinate of the new object (e.g., 0.03mm).
	<*Z*>	Integer	Y coordinate of the new object (e.g., 0mm).
Return Value	Returns the name of the new 3D line.		

Python Syntax	AddObject(<ObjectType>, [<Parameters>])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example

```
oEditor.AddObject("3D Line",
  [
    "NAME:Contents",
    "3D LineGeometry:=",
    [
       "Name:=", "3D Line_12904",
      "LayerName:=", "Postprocessing",
      "lw:=" , "0",
      "gm:=",
      [
        "m:=", 1,
        "pnt:=", [ "lid:=", 2, "pos:=", [ "X:=", "-0.08mm", "Y:=", "0.03mm",
        "Z:=", "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" ,[ "X:=" , "-0.07mm", "Y:=" , "0.03mm",
         "Z:=" , "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" , [ "X:=" , "-0.07mm", "Y:=" ,
        "0.021mm", "Z:=" , "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" , [ "X:=" , "-0.08mm", "Y:=" ,
        "0.02mm", "Z:=" , "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" , [ "X:=" , "-0.08mm", "Y:=" ,
        "0.03mm", "Z:=" , "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" , [ "X:=" , "-0.08mm", "Y:=" ,
        "0.03mm", "Z:=" , "0"]],
        "pnt:=" , [ "lid:=" , 2, "pos:=" , [ "X:=" , "-0.08mm", "Y:=" ,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"0.03mm", "Z:=" , "0"]]]]])

CreateCircle (Layout Editor)

Creates a circle primitive object and adds it to the current layout.

UI Access	None
Parameters	See table below
Return Value	Object name (string)

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*CircleGeometry*>	Array	Array of geometric properties for the circle.
<*LayerName*>	String	Name of the layer.
<*Lw*>	Integer	Border line width.
<*X*>	Integer	Center X coordinate.
<*Y*>	Integer	Center Y coordinate.
<*R*>	Integer	Radius.

Python Syntax	CreateCircle([<*NAME:Contents*>, <*CircleGeometry*>, [<*LayerName*>, <*Lw*>, <*X*>, <*Y*>, <*R*>]])
Python Example	oEditor.CreateCircle([

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:Contents",
        "CircleGeometry:=", [
        "LayerName:=", "TopLayer",
        "Lw:=", "0.2mm",
        "X:=", "10mm",
        "Y:=", "15mm",
        "R:=", "5mm"
      ]
    ]
)
```

CreateLine (Layout Editor)

Creates a line primitive object.

UI Access	None

Parameters

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*LineGeometry*>	Array	Array of geometric properties for the line.
<*LayerName*>	String	Name of the layer.
<*Lw*>	Integer	Line width.
<*EndStyle*>	Integer	Line end style: 0 = flat end, 1 = extended end, 2 = round end.
<*JoinStyle*>	Integer	Line join style: 0 = unmitered join, 1 = mitered join, 2 = optimally mitered join, 3 = radially mitered join, 4 = chord mitered join.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*VertexSequence*>	Array	Sequence of vertex coordinates: "n:=" count, followed by "x0:=", "y0:=", "x1:=", "y1:=", etc.
Return Value	Object name (string)		

Python Syntax	CreateLine([<*NAME:Contents*>, <*LineGeometry*>, [<*LayerName*>, <*Lw*>, <*EndStyle*>, <*JoinStyle*>, <*VertexSequence*>]])

Python Example	`oEditor.CreateLine(`
	`  [`
	`  "NAME:Contents",`
	`  "LineGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "Lw:=", 2,`
	`  "EndStyle:=", 1,`
	`  "JoinStyle:=", 2,`
	`  "n:=", 2,`
	`  "x0:=", 10,`
	`  "y0:=", 15,`
	`  "x1:=", 25,`
	`  "y1:=", 30`
	`  ]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

AddObject ("Partition Line") (IC Editor)

Create an additional partition line for a mesh region (i.e., CreateMeshRegion).

UI Access	From the **Layout Editor**, select a mesh region. Then right-click and select **Mesh Region** > **Add a partition line**.
Parameters	See table below

Name	Type	Description
<ObjectType>	String	Type of object to add (i.e., must have the value "mesh region partition line").
<Contents>	Array	Structured array containing these parameters in order: • <NAME:Contents> • <owner> • <mesh region partition lineGeometry> • <snap start> • <snap end>
<NAME:Contents>	String	Identifies the contents of the array (i.e., "Contents").
<owner>	String	Name of the mesh region in which to create the new partition line.
<mesh region partition lineGeometry>	Array	Name of the array containing the geometry of the new partition line, containing these parameters in order: • <Name> • <LayerName> • <lw> • <endstyle>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <StartCap> • <EndCap> • <n> • <U> • <x> • <y> • <MR>
<Name>	String	Name of the new partition line.
<LayerName>	Integer	Name of the layer on which to create the new partition line (i.e., "MeshRegions").
<lw>	Integer	Line width of the new partition line, including unit of measurement (i.e., "0.01mm").
<endstyle>	Integer	Of the new partition line, all vertices (i.e., points) will be: **0** — round. **1** — corner (i.e., square). **2** — mitered (i.e., extended to a point where it meets a corner).
<StartCap>	Integer	Of the new partition line, the starting point of the line will be: **0** — round. **1** — flat. **2** — extended (i.e., extended from the starting point beyond the actual beginning of its path).
<Endcap>	Integer	Of the new partition line, the ending point of the line will be: **0** — round. **1** — flat. **2** — extended (i.e., extended beyond the actual end point). **3** — Same as Start.
<n>	Integer	The number of vertices in the new partition line.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<U>	String	The unit of measurement by which the new partition will be created.
	<x>	Integer	X coordinate of a point in the new partition line.
	<y>	Integer	Y coordinate of a point in the new partition line.
	<MR>	String	A number and unit of measurement representing the miter ratio (i.e., of the new partition line, assuming multiple segments, the size of the point where two segments meet).
	<snap start>	Boolean	**True** start of the new partition line will snap to a border of the mesh region.
	<snap end>	Boolean	**True** end of the new partition line will snap to a border of the mesh region.
Return Value	Returns the name of the new partition line.		

Python Syntax	AddObject(<ObjectType>, [<Contents>])
Python Example	oEditor.AddObject("mesh region partition line", ["NAME:Contents", "owner:=" , "meshregion__11", "mesh region partition lineGeometry:=", ["Name:=" , "line__18", "LayerName:=" , "MeshRegions", "lw:=" , "0.01mm", "endstyle:=" , 0, "StartCap:=" , 0, "EndCap:=" , 0, "n:=" , 3,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "U:=" , "mm",
        "x:=" , 19,
        "y:=" , 9,
        "x:=" , 15,
        "y:=" , 4,
        "x:=" , 20,
        "y:=" , 1,
        "MR:=" , "600mm"],
    "snap start:=" , True,
    "snap end:=" , True
])
```

CreatePolygon (Layout Editor)

Creates a polygon primitive object.

UI Access	None
Parameters	

Name	Type	Description
<NAME:Contents>	String	Object descriptor header.
<LineGeometry>	Array	Array of geometric properties for the polygon, consisting of the following parameters in order:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td></td><td>
<table>
<tr><td></td><td></td><td>

- *<LayerName>*
- *<Lw>*
- *<N>*
- *<X0>*
- *<Y0>*
- *<X1>*
- *<Y1>*
- *<X2>*
- *<Y2>*
- *<X3>*
- *<Y3>*

</td></tr>
<tr><td>*<LayerName>*</td><td>String</td><td>Name of the layer.</td></tr>
<tr><td>*<Lw>*</td><td>Integer</td><td>Border line width.</td></tr>
<tr><td>*<N>*</td><td>Integer</td><td>Number of polygon vertices.</td></tr>
<tr><td>*<X0>*</td><td>Double</td><td>X-coordinate of vertex 0.</td></tr>
<tr><td>*<Y0>*</td><td>Double</td><td>Y-coordinate of vertex 0.</td></tr>
<tr><td>*<X1>*</td><td>Double</td><td>X-coordinate of vertex 1.</td></tr>
<tr><td>*<Y1>*</td><td>Double</td><td>Y-coordinate of vertex 1.</td></tr>
<tr><td>*<X2>*</td><td>Double</td><td>X-coordinate of vertex 2.</td></tr>
<tr><td>*<Y2>*</td><td>Double</td><td>Y-coordinate of vertex 2.</td></tr>
<tr><td>*<X3>*</td><td>Double</td><td>X-coordinate of vertex 3.</td></tr>
<tr><td>*<Y3>*</td><td>Double</td><td>Y-coordinate of vertex 3.</td></tr>
</table>
</td></tr>
<tr><td>**Return Value**</td><td>Object name (string)</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreatePolygon([<*NAME:Contents*>, [<*LineGeometry*>]])

Python Example	oEditor.CreatePolygon(
	  [
	  "NAME:Contents",
	  "LineGeometry:=", [
	  "LayerName:=", "TopLayer",
	  "Lw:=", 2,
	  "n:=", 4,
	  "x0:=", 10,
	  "y0:=", 15,
	  "x1:=", 25,
	  "y1:=", 30,
	  "x2:=", 40,
	  "y2:=", 20,
	  "x3:=", 15,
	  "y3:=", 5
	  ]
	  ]
	)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateRectangle (Layout Editor)

Creates a rectangle primitive object and adds it to the current layout.

UI Access	None
Parameters	

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*RectGeometry*>	Array	Array of geometric properties for the rectangle.
<*LayerName*>	String	Name of the placement layer.
<*Lw*>	Integer	Line width.
<*X*>	Integer	Center X coordinate.
<*Y*>	Integer	Center Y coordinate.
<*W*>	Integer	Width (X-direction size).
<*H*>	Integer	Height (Y-direction size).
<*Ang*>	Integer	Rotation angle.

Return Value	Object name (string)

Python Syntax	CreateRectangle([<*NAME:Contents*>, <*RectGeometry*>, [<*LayerName*>, <*Lw*>, <*X*>, <*Y*>, <*W*>, <*H*>, <*Ang*>]])
Python Example	

```
oEditor.CreateRectangle(
  [
    "NAME:Contents",
    "RectGeometry:=", [
      "LayerName:=", "TopLayer",
      "Lw:=", 2,
      "X:=", 10,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Y:=", 15,
        "W:=", 30,
        "H:=", 20,
        "Ang:=", 45
      ]
    ]
)
```

CreateText (Layout Editor)

Creates a text primitive object and adds it to the current layout.

UI Access	None

Parameters

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*TextGeometry*>	Array	Array of geometric and formatting properties for the text.
<*LayerName*>	String	Name of the placement layer.
<*X*>	Integer	Origin X coordinate.
<*Y*>	Integer	Origin Y coordinate.
<*Ang*>	Integer	Rotation angle.
<*IsPlot*>	Boolean	Whether the text is plotted.
<*Font*>	String	Font name.
<*Size*>	Integer	Font size.
<*Weight*>	Integer	Text weight.
<*Just*>	Integer	Text justification.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Mirror>	Boolean	Whether the text is mirrored.
	<Text>	String	Text content to display.
Return Value	Object name (string)		

Python Syntax	CreateText([*<NAME:Contents>*, *<TextGeometry>*, [*<LayerName>*, *<X>*, *<Y>*, *<Ang>*, *<IsPlot>*, *<Font>*, *<Size>*, *<Weight>*, *<Just>*, *<Mirror>*, *<Text>*]])

Python Example	`oEditor.CreateText(`
	`  [`
	`  "NAME:Contents",`
	`  "TextGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "X:=", 10,`
	`  "Y:=", 15,`
	`  "Ang:=", 0,`
	`  "IsPlot:=", true,`
	`  "Font:=", "Arial",`
	`  "Size:=", 12,`
	`  "Weight:=", 3,`
	`  "Just:=", 4,`
	`  "Mirror:=", false,`
	`  "Text:=", "Hello World"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]])

Create Voids in Primitives

The following primitives are available.

CreateCircleVoid

CreateLineVoid

CreatePolygonVoid

CreateRectangleVoid

GetPolygonVoids

CreateCircleVoid (Layout Editor)

Creates a circle void and adds it to a particular parent primitive.

UI Access	Available only through scripting.
Parameters	

Name	Type	Description
<NAME:Contents>	String	Header string identifying the array contents.
<Owner>	String	Name of the parent primitive object to which the void is added.
<Circle VoidGeo-metry>	Array	Contains geometric parameters for the void. Format is assumed to match `CreateCircle` geometry array.
<*LayerName*>	String	Name of the layer.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Lw>	Integer	Border line width.
	<X>	Integer	Center X coordinate.
	<Y>	Integer	Center Y coordinate.
	<R>	Integer	Radius
Return Value	Void object name.		

Python Syntax	oEditor.CreateCircleVoid([<NAME:Contents>, <Owner>, <Circle VoidGeometry>])

Python Example	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.CreateCircleVoid(`
	`  ["NAME:Contents",`
	`  "Owner:=", "Pad1",`
	`  "Circle VoidGeometry:=",`
	`  ["LayerName:=", "Top",`
	`  "lw:=", "0.1mm",`
	`  "x:=", "0mm",`
	`  "y:=", "0mm",`
	`  "r:=", "0.5mm"]`
	`  ]`
	`)`

CreateLineVoid (Layout Editor)

Creates a line void and adds it to a specified parent primitive.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Object descriptor header.</td></tr>
<tr><td><Owner></td><td>String</td><td>Name of the parent primitive.</td></tr>
<tr><td><Line voidGeometry></td><td>Array</td><td>Array defining the line void geometry by two pairs of X,Y coordinates.</td></tr>
<tr><td><X1></td><td>Integer</td><td>Start X coordinate.</td></tr>
<tr><td><Y1></td><td>Integer</td><td>Start Y coordinate.</td></tr>
<tr><td><X2></td><td>Integer</td><td>End X coordinate.</td></tr>
<tr><td><Y2></td><td>Integer</td><td>End Y coordinate.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Void object name.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>CreateLineVoid([<NAME:Contents>, <Owner>, <Line VoidGeometry>, [<X1>, <Y1>, <X2>, <Y2>]])</td></tr>
<tr><td>Python Example</td><td>

```
oEditor.CreateLineVoid(
  [
       "NAME:Contents",
       "Line VoidGeometry:=", [
       "X1:=", "0mm",
       "Y1:=", "0mm",
       "X2:=", "10mm",
       "Y2:=", "0mm"
    ]
```

</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreatePolygonVoid (Layout Editor)

Creates a polygon void and adds it to a specified parent primitive.

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME:Contents*></td><td>String</td><td>Object descriptor header.</td></tr><tr><td><*Owner*></td><td>String</td><td>Name of the parent primitive.</td></tr><tr><td><*Poly VoidGeometry*></td><td>Array</td><td>Array of X,Y coordinate pairs defining the polygon void.</td></tr><tr><td><*X*></td><td>Integer</td><td>X coordinate of a polygon vertex.</td></tr><tr><td><*Y*></td><td>Integer</td><td>Y coordinate of a polygon vertex.</td></tr></table>
Return Value	Void object name.

Python Syntax	CreatePolygonVoid([<*NAME:Contents*>, <*Owner*>, <*Poly VoidGeometry*>, [<*X*>, <*Y*>, ...]])

Python Example	`oEditor.CreatePolygonVoid(`
	`  [`
	`  "NAME:Contents",`
	`  "Owner:=", "TopLayer",`
	`  "Poly VoidGeometry:=", [`
	`  "X:=", "0mm",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Y:=", "0mm",
        "X:=", "10mm",
        "Y:=", "0mm",
        "X:=", "10mm",
        "Y:=", "10mm",
        "X:=", "0mm",
        "Y:=", "10mm"
      ]
   ]
)
```

CreateRectangleVoid (Layout Editor)

Creates a rectangle void and adds it to a specified parent primitive.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.
	<Owner>	String	Name of the parent primitive.
	<Rect VoidGeometry>	Array	Array defining the rectangle void geometry by two pairs of X,Y coordinates.
	<X1>	Integer	Start X coordinate.
	<Y1>	Integer	Start Y coordinate.
	<X2>	Integer	End X coordinate.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Y2>	Integer	End Y coordinate.
Return Value	Void object name.		

Python Syntax	CreateRectangleVoid([*<NAME:Contents>*, *<Owner>*, *<Rect VoidGeometry>*, [*<X1>*, *<Y1>*, *<X2>*, *<Y2>*]])

Python Example	`oEditor.CreateRectangleVoid(`
	`  [`
	`  "NAME:Contents",`
	`  "Owner:=", "TopLayer",`
	`  "Rect VoidGeometry:=", [`
	`  "X1:=", "0mm",`
	`  "Y1:=", "0mm",`
	`  "X2:=", "10mm",`
	`  "Y2:=", "5mm"`
	`  ]`
	`  ]`
	`)`

GetPolygonVoids (Layout Editor)

Returns the names of the voids inside the selected polygon.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PolyName*></td><td>String</td><td>Target name of a polygon in the active design.</td></tr></table>
Return Value	Array of the names of the polygon voids inside the target polygon, separated by commas, or None if no voids exist.

Python Syntax	GetPolygonVoids (<PolyName>)
Python Example	`oEditor.GetPolygonVoids('rect_0')` `['circle void_1', 'circle void_2', 'circle_void 3']`

Other Creation Methods

The following creation methods are available.

AddPinGroupRefPort

CreateComponent

CreateCurrentSourcePort

CreateHole

CreateMeasure

CreatePin

CreatePinGroup

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreatePinGroupPort

CreatePinGroups

CreateTrace

CreateVia

CreateViaGroups

CreateVoltageProbePort

CreateVoltageSourcePort

DeletePinGroup

AddPinGroupRefPort (Layout Editor)

Assigns a reference port to a pin group.

> **Note:**
>
> AddPinGroupRefPort is also called when the command CreatePinGroupPort (Layout Editor) is used.

UI Access	From **Layout**, select **Draw Ports/Sources**.
Parameters	

Name	Type	Description
<PinGroup>	Array	Array consisting of the name of a pingroup that requires a reference port (e.g., PinGroup_2).
<PortRef>	Array	Array consisting of the name of a pingroup that acts as reference port (e.g., PinGroup_1).

Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddPinGroupRefPort ([<PinGroup>], [<PortRef>])
Python Example	`oEditor.AddPinGroupRefPort(["PinGroup_2"], ["PinGroup_1"])`

CreateComponent (Layout Editor)

Places a component in the layout editor.

UI Access	None
Parameters	

Name	Type	Description
<*NAME:Contents*>	String	Required header string identifying the parameter block.
<*Definition_Name*>	String	Name of the component definition to place.
<*Placement*>	Array	Structured array containing: `X`, `Y`
<*X*>	Double	X coordinate of the component origin.
<*Y*>	Double	Y coordinate of the component origin.
<*Layer*>	String	Name of the layer on which to place the component.
<*StackupLayers*>	Array	Optional array of layer mappings. Each entry contains: `FootprintLayer`, `DesignLayer`
<*FootprintLayer*>	String	Name of the footprint stackup layer.
<*DesignLayer*>	String	Name of the corresponding design stackup layer.
<*DrawLayers*>	Array	Optional array of drawing layer mappings. Each entry contains: `FootprintLayer`, `DesignLayer`
<*FootprintLayer*>	String	Name of the footprint drawing layer.
<*DesignLayer*>	String	Name of the corresponding design drawing layer.

Return Value	Returns the name of the newly created component.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note: Each Layer mapping is specified as a single quoted string in the format "<footprint layer>:<design layer>". If a layer does not map to anything, use "0" for the unmapped side (e.g., "Measures:0" — footprint layer "Measures" has no design layer mapping, or "0:Dielectric" — design layer "Dielectric" has no footprint layer mapping.

Note: The StackupLayers and DrawLayers arguments are optional. If omitted, mappings are determined automatically.

Python Syntax	CreateComponent([<NAME:Contents>, <Definition_Name>, <Placement>, <Layer>, <StackupLayers>, <DrawLayers>])
Python Example	oEditor.CreateComponent(["NAME:Contents", "Definition_Name:=", "MyComponent", "Placement:=", ["X:=", 0.0, "Y:=", 0.0], "Layer:=", "TopLayer", "StackupLayers:=", ["Top:Dielectric", "Measures:0"], "DrawLayers:=", ["Silkscreen:Top", "0:Dielectric"]])

CreateCurrentSourcePort (Layout Editor)

Create a current source port and enable placement on the **Layout Editor**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Draw** > **Port**.		
Parameters	Name	Type	Description
	<NAME:Location>	String	Required first parameter which designates the contents of the array (i.e., the Location of the new port).
	<PosLayer>	String	Name of the stackup layer on which to place the positive end of the new port (e.g., Trace).
	<X0>	Integer	X coordinate of the positive end of the new port.
	<Y0>	Integer	Y coordinate of the positive end of the new port.
	<NegLayer>	String	Name of the stackup layer on which to place the negative end of the new port.
	<X0>	Integer	X coordinate of the negative end of the new port.
	<Y0>	Integer	Y coordinate of the negative end of the new port.
Return Value	None		

Python Syntax	CreateCurrentSourcePort([<NAME:Location>, <PosLayer>, <X0>, <Y0>, <NegLayer>, <X1>, <Y1>])

Python Example	`oEditor.CreateCurrentSourcePort`
	`(["NAME:Location", "PosLayer:=" , "Trace", "X0:=" , -0.011, "Y0:=" , 0.004,`
	`"NegLayer:=" , "Trace", "X1:=" , -0.011, "Y1:=" , 0.004)`

CreateHole (Layout Editor)

Places a Hole object.

UI Access	From **Layout Editor**, navigate to **Draw** and select **Hole**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<Contents>	Array	Structured array containing the full hole definition and all associated parameters.
<NAME:Full_Definition>	Array	Structured array containing hole type, properties, layer range, and placement.
<Type>	String	Must be set to "hole".
<Properties>	Array	Structured array of hole geometry parameters: <VariableProp> entries for radius and sides.
<VariableProp>	Array	Each entry defines a named property: ("Radius", "D", "", <Value>) or ("Sides", "D", "", <iInteger>).
<From_Layer>	String	Name of the starting layer for the hole.
<To_Layer>	String	Name of the ending layer for the hole.
<Placement>	Array	Structured array defining the hole position: <X>, <Y> coordinates.
<X>	Integer	X coordinate of the hole center.
<Y>	Integer	Y coordinate of the hole center.
<Layer>	String	Layer name used for placement reference.

Return Value

Returns the name of the newly created Hole.

Python Syntax

CreateHole([<Contents>])

Python Example

```
oEditor.CreateHole([
  "NAME:Contents",
  ["NAME:Full_Definition",
    "Type:=", "hole",
    ["NAME:Properties",
      "VariableProp:=", ["radius", "D", "", 1.5],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "VariableProp:=", ["sides", "D", "", 6]],
    "From_Layer:=", "Top",
    "To_Layer:=", "Bottom",
    "Placement:=", ["x:=", 10, "y:=", 20],
    "Layer:=", "Top"]
])
```

CreateMeasure (Layout Editor)

Creates a measurement.

UI Access	None.		
Parameters	*<Measure_Array>*	Array	An array consisting of the following parameters in order: • *<NAME:Contents>* • *<MeasurementGeometry>* • *<text_style>*
	<NAME:Contents>	String	Identifies the contents of the measurement definition array.
	<Meas-urementGeometry>	Array	Array containing the measurement geometry parameters in order: • *<LayerName>* • *<Lw>* • *<Sx>* • *<Sy>*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• *<Ex>* • *<Ey>*
<LayerName>	String	Specifies the layer on which the measurement geometry is drawn.
<Lw>	Double	Line width of the measurement geometry.
<Sx>	Double	X-coordinate of the measurement start point.
<Sy>	Double	Y-coordinate of the measurement start point.
<Ex>	Double	X-coordinate of the measurement end point.
<Ey>	Double	Y-coordinate of the measurement end point.
<text_style>	Array	Text style parameters in order: • *<Name>* • *<IsPlot>* • *<Font>* • *<Size>* • *<Angle>* • *<Weight>* • *<Just>* • *<Mirror>* • *<Scales>*
<Name>	String	Name assigned to the measurement text object.
<IsPlot>	Boolean	Indicates whether the measurement text is plotted.
<Font>	String	Font used for the measurement text.
<Size>	Double	Font size of the measurement text.
<Angle>	Double	Rotation angle of the text in degrees.
<Weight>	String	Text weight (i.e., Normal, Bold).
<Just>	String	Text justification (i.e., Left, Center, Right).
<Mirror>	Boolean	Mirrors the text horizontally when True.
<Scales>	Boolean	Determines whether the text scales with zoom level.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Name of the measured object.

Python Syntax	oEditor.CreateMeasure([<NAME:Contents>, <MeasurementGeometry>, <text_style>])

Python Example	`oEditor.CreateMeasure(`
	`  [`
	`  "NAME:Contents",`
	`  "MeasurementGeometry:=", ["LayerName:=", "TopLayer", "lw:=", 1.0, "sx:=", 0.0, "sy:=", 0.0, "ex:=", 10.0, "ey:=", 10.0],`
	`  "text_style:=", ["name:=", "MyMeasure", "isPlot:=", True, "font:=", "Arial", "size:=", 12, "angle:=", 0, "weight:=", "Normal", "just:=", "Left", "mirror:=", False, "scales:=", True]`
	`])`

CreatePin (Layout Editor)

Creates a pin.

UI Access	None.
Parameters	See table below.

Name	Type	Description
<Pin_Description>	Array	Array consisting of the following parameters in order: • *<NAME:Contents>*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• *<Port>* • *<Rotation>* • *<Offset>* • *<Padstack>*
	<NAME:Contents>	String	Object descriptor header.
	<Port>	Array	Port definition array (not used).
	<Name>	String	Object name (unused).
	<Rotation>	Array	Rotation angle of the pin.
	<Value>	Double	Rotation angle value.
	<Offset>	VPoint	Pin position.
	<Padstack>	String	Padstack name.
Return Value	Pin name		

Python Syntax	oEditor.CreatePin(<Pin_Description>)

Python Example	`oEditor.CreatePin(`
	`  [`
	`  "NAME:Contents",`
	`  ["NAME:Port", "Name:=", "MyObject"],`
	`  "Rotation:=", [90],`
	`  "Offset:=", [0.0, 0.0],`
	`  "Padstack:=", "MyPadstack"`
	`  ]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreatePinGroup (Layout Editor)

Use: Create a pin group from a selection of pins.

Syntax: **CreatePinGroup(STRING *pin-group-name*, ["NAME:elements", STRING *pin*, STRING *pin*, …])**

CreatePinGroupPort (Layout Editor)

Use: Create a pin group port from two pin-groups; first is positive, second is negative.

Syntax: **CreatePinGroupPort(["Name:elements", STRING *pos-pin-group*, STRING *neg-pin-group*])**

CreatePinGroups (Layout Editor)

Creates pin groups from a selection of pins.

<table>
<tr><td>UI Access</td><td>After selecting two or more pins, right-click within the Layout Editor and select Port > Create Pin Group.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:PinGroupDatas>
<PinGroup></td></tr>
<tr><td><NAME:PinGroupDatas></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "PinGroupDatas").</td></tr>
<tr><td><PinGroup></td><td>Array</td><td>Structured array containing these parameters in order, followed by <PinGroup2>, <PinGroup3>, etcetera:
<NAME:>
<PinNames></td></tr>
<tr><td><NAME:></td><td>String</td><td>Required first parameter which must have the name of the pin group (e.g., "PinGroup_3").</td></tr>
<tr><td><PinNames></td><td>String</td><td>Names of the pins to group, separated by commas.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	CreatePinGroups([<NAME:PinGroupDatas>, [<NAME:>, <PinNames>]])

Python Example	`oEditor.CreatePinGroups(`
	`[`
	`  "NAME:PinGroupDatas",`
	`  ["NAME:PinGroup_3",`
	`  "via_2",`
	`  "via_1"`
	`  ]`
	`])`

CreateTrace (Layout Editor)

Creates a trace with a manually specified path between pins, ports, or selected edges.

UI Access	From **Layout Editor**, navigate to **Draw** and select **Route** > **Manual**.
Parameters	

Name	Type	Description
<Options>	Array	A structured array containing layer pairings for routing: <Layer1>, <Layer2>.
<Layer1>	String	Name of the starting layer.
<Layer2>	String	Name of the ending layer.
<Lines>	Array	-A structured array defining one or more trace segments: <Num>, <Line Name>, <Layer Name>, <Line Width>, <End

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		Style>, <Join Style>, <Num Vertices>, <Units>, <X>, <Y>.
<Num>	Integer	Number of line entries.
<Line Name>	String	Name of the trace line.
<Layer Name>	String	Layer the trace is drawn on.
<Line Width>	String	Width of the trace, including units.
<End Style>	Integer	Cap style for line ends.
<Join Style>	Integer	Corner join style.
<Num Vertices>	Integer	Number of vertex points.
<Units>	String	Units for coordinates (e.g., "mm").
<X>, <Y>	Integer	Vertex coordinates.
<Vias>	Array	A structured array defining via instances and geometry: <Num>, <Via Name>, <Referenced Padstack>, <X>, <Y>, <Rotation>, <Override Hole>, <Hole Diameter>, <Highest Layer>, <Lowest Layer>.
<Num>	Integer	Number of via entries.
<Via Name>	String	Name of the via.
<Referenced Pad-stack>	String	Padstack definition name.
<X>, <Y>	Integer	Coordinates for via placement.
<Rotation>	String	Via orientation angle, including units (e.g., "180deg").
<Override Hole>	Boolean	If true, overrides the default hole diameter.
<Hole Diameter>	String	Override diameter value, including units.
<Highest Layer>	String	Top layer the via connects to.
<Lowest Layer>	String	Bottom layer the via connects to.
<Elements>	Array	A structured array listing ports or pins to connect: <Port Name>.
<Port Name>	String	Name of the port or pin.
<EdgeRefs>	Array	A structured array referencing primitive or via edges for trace attachment: <Primitive Name>, <Edge Number>, <Via Name>, <Layer ID>, <Start X>, <Start Y>, <End X>, <End Y>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<Arc Height>, <Radians>.
	<Primitive Name>	String	Name of the primitive containing the edge.
	<Edge Number>	Integer	Index of the edge on the primitive.
	<Via Name>	String	Name of the via used for arc reference.
	<Layer ID>	Integer	ID of the layer containing the arc.
	<Start X>, <Start Y>	Integer	Coordinates of the arc start point.
	<End X>, <End Y>	Integer	Coordinates of the arc end point.
	<Arc Height>	Integer	Height of the arc (0 for straight edge).
	<Radians>	Integer	Arc size in radians (0 for straight edge).
Return Value	None		

Python Syntax	CreateTrace([[<Options>], [<Lines>], [<Vias>], [<Elements>], [<EdgeRefs>]])
Python Example	oEditor.CreateTrace([["NAME:Options", "Layer1:=", "Top", "Layer2:=", "Bottom"], ["NAME:Lines", "Num:=", 1, "l0:=", ["Name:=", "Trace1", "LayerName:=", "Top", "Lw:=", "1mm", "EndStyle:=", 0, "JoinStyle:=", 1, "N:=", 2, "U:=", "mm", "X:=", 10, "Y:=", 20]], ["NAME:Vias", "Num:=", 1, "NAME:V1", "Name:=", "Via1", "ReferencedPadstack:=", "Pad1", "VPosition:=", ["X:=", 10, "Y:=", 20], "VRotation:=", ["180deg"], "Overrides Hole:=", True, "Hole Diameter:=", ["1mm"], "Highest Layer:=", "Top", "Lowest Layer:=", "Bottom"], ["NAME:Elements", "Port1"], ["NAME:EdgeRefs", "Edge:=", ["Et:=", "Pe", "Prim:=", "Prim1", "Edge:=", 3]]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateVia (Layout Editor)

Use: Creates a new via.

Syntax: **CreateVia** <via_description>

Return Value: Returns the name of the created via

Parameters: <via_description>:

```
Array("NAME:Contents",
"name:=", <object_name>, // not used
"vposition:=", <vpoint>, // position
"rotation:=", double, // rotation in radians
"overrides hole:=", <bool>,// overrides or not padstack hole
"hole diameter:=", Array(<value>), // via diameter
"ReferencedPadstack:=", <quoted_string>, // padstack name
"highest_layer:=", <layer_name>,
"lowest_layer:=", <layer_name>)
```

CreateViaGroups (Layout Editor)

Creates via groups on the target layer.

<table>
<tr><td>UI Access</td><td colspan="3">After selecting two or more vias, navigate to Draw > Via Groups > Create > Persistent/Non-Persistent.</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:</td></tr>
</table></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <NAME:Vias> • One or more <TargetObjects> • <GroupName> • <Enable>
	<NAME:vias>	String	Required literal string `"NAME:Vias"`. Must appear first.
	<TargetObjects>	String	Names of via primitives to include in the group. Provide one string per via, separated by commas.
	<GroupName>	String	Names of the new via group.
	<Settings>	Array	Structured array containing these parameters in order: • <NAME:Setting Contents> • <clustering_method> • <via_grouping_tolerance>
	<NAME:Setting Contents>	String	Required literal string `"NAME:Setting Contents"`. Must be the first element of the settings array that follows the via list. Introduces the settings block for clustering parameters.
	<clustering_method>	String	Clustering algorithm for grouping vias. • **"Proximity"** — spatial proximity clustering • **"Range"** — distance-based range clustering • **"Density"** — HDBSCAN density-based clustering Default: `"Proximity"`
	<via_grouping_tolerance>	String	Distance tolerance used for **Proximity** and **Range** clustering. Specify any valid layout distance unit (e.g., `"50um"`, `"1mm"`). Ignored when `clustering_method="Density"`.
Return Value	None.		

Python Syntax	CreateViaGroups([<NAME:Vias>, <TargetObjects...>, <GroupName>], [<NAME:Setting Contents>, <clus-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	tering_method>, <via_grouping_tolerance>])
Python Example	oEditor.CreateViaGroups(["NAME:Vias", "circle_8651", "rect_825", "ViaGroup_1"], ["NAME:Setting Contents", "clustering_method:=", "Proximity", "via_grouping_tolerance:=", "100um"])

CreateVoltageProbePort (Layout Editor)

Create a voltage probe port and enable placement on the **Layout Editor**.

UI Access	**Draw** > **Port**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><NAME:Location></td><td>String</td><td>Required first parameter which designates the contents of the array (i.e., the Location of the new port).</td></tr><tr><td><PosLayer></td><td>String</td><td>Name of the stackup layer on which to place the positive end of the new port (e.g., Trace).</td></tr><tr><td><X0></td><td>Integer</td><td>X coordinate of the positive end of the new port.</td></tr><tr><td><Y0></td><td>Integer</td><td>Y coordinate of the positive end of the new port.</td></tr><tr><td><NegLayer></td><td>String</td><td>Name of the stackup layer on which to place the negative end of the new port.</td></tr><tr><td><X0></td><td>Integer</td><td>X coordinate of the negative end of the new port.</td></tr><tr><td><Y0></td><td>Integer</td><td>Y coordinate of the negative end of the new port.</td></tr></table>
Return Value	String PortID, the identifier that can be used for this port in script operations involving this schematic

Python Syntax	CreateVoltageProbePort([<NAME:Location>, <PosLayer>, <X0>, <Y0>, <NegLayer>, <X1>, <Y1>])

Python Example	`oEditor.CreateVoltageProbePort`
	`(["NAME:Location", "PosLayer:=" , "Trace", "X0:=" , -0.011, "Y0:=" , 0.004,`
	`"NegLayer:=" , "Trace", "X1:=" , -0.011, "Y1:=" , 0.004)`

CreateVoltageSourcePort (Layout Editor)

Create a voltage source port and enable placement on the **Layout Editor**.

UI Access	**Draw** > **Port**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><NAME:Location></td><td>String</td><td>Required first parameter which designates the contents of the array (i.e., the</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Location of the new port).
	<PosLayer>	String	Name of the stackup layer on which to place the positive end of the new port (e.g., Trace).
	<X0>	Integer	X coordinate of the positive end of the new port.
	<Y0>	Integer	Y coordinate of the positive end of the new port.
	<NegLayer>	String	Name of the stackup layer on which to place the negative end of the new port.
	<X0>	Integer	X coordinate of the negative end of the new port.
	<Y0>	Integer	Y coordinate of the negative end of the new port.
Return Value	None		

Python Syntax	CreateVoltageSourcePort([<NAME:Location>, <PosLayer>, <X0>, <Y0>, <NegLayer>, <X1>, <Y1>])

Python Example	`oEditor.CreateVoltageSourcePort`
	`(["NAME:Location", "PosLayer:=" , "Trace", "X0:=" , -0.011, "Y0:=" , 0.004,`
	`"NegLayer:=" , "Trace", "X1:=" , -0.011, "Y1:=" , 0.004)`

DeletePinGroup (Layout Editor)

Use: Deletes a pin group.

Syntax: **DeletePinGroup(STRING *pin-group-name*)**

Object Movement and Modification Methods

The documented object movement methods are available.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Connect (Layout Editor)

Copy (Layout Editor)

Delete (Layout Editor)

Disconnect (Layout Editor)

Edit (Layout Editor)

Expand (Layout Editor)

ExpandWithPorts (Layout Editor)

FlipHorizontal (Layout Editor)

FlipVertical (Layout Editor)

Move (Layout Editor)

Paste (Layout Editor)

Rotate (Layout Editor)

Connect (Layout Editor)

Connects the referenced objects. After execution, the objects will belong to the same electrical net.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:elements>	String	Required first parameter which must have the specified value (i.e., "elements").
	<object_name>	String	Name of the object to connect. Multiple objects may be listed.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Connect(<*NAME:elements*>, <*object_name*>, <*object_name*>, …)

Python Example	`oEditor.Connect(`
	`  ["NAME:Elements",`
	`  "rect_01",`
	`  "rect_02"`
	`  ])`

Copy (Layout Editor)

Use: Copies the referenced objects to the clipboard.

Syntax: **Copy** Array (<object_name>, // 1st object

`<object_name>, // 2nd object`

Return Value: Returns the name of the new net.

Delete (Layout Editor)

Use: Deletes one or more objects.

Syntax: **Delete** <object_name_list>// names of the objects to be deleted

`<object_name_list>:`

`Array ("NAME:elements", <object_name>, <object_name>, …)`

Disconnect (Layout Editor)

Use: Removes the specified object(s) from their current electrical net(s) (if any).

Syntax: **Disconnect** Array ("NAME: *elements*",

`<object_name>, // 1st object`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<object_name>, // 2nd object, if any

…) // etc

Edit (Layout Editor)

Use: Causes modification of one or more existing object(s).

Syntax: **Edit** <Array("NAME:items".....)>,

Parameters: <Array("NAME:items"
<edit_object_info>, // one object info

<edit_object_info>, // another one

…) // etc

<edit_object_info>:

Array("NAME:item",

"name:=", <object_name>, // name of the object

<object_description>) // 'new' object state, type should be consistent with <object_name>:

<object_description>:

<circle_description> |

<rectangle_description> |

<line_description> |

<polyon_description> |

<text_description> |

<circle_void_description> |

<rectangle_void_description> |

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<line_void_description> |
<polyon_void_description> |
<component_description> |
<pin_description> |
<via_description>
```

Expand (Layout Editor)

Scales an existing circle, square, or polygon. To expand an object while preserving ports, use ExpandWithPorts.

UI Access	**Draw** > **Expand**.		
Parameters	Name	Type	Description
	<expansion>	String	Expansion amount, including unit. Value can be negative to shrink an object.
	<join>	String	Corner style. One of: • "ROUND" • "MITER" • "CORNER"
	<replace>	Boolean	True to replace original object; False to create new object.

	<items>	Array	Structured array containing object to expand:
			`["NAME:elements",`
			`"<Name of Object to Expand>"]`

Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Expand (<*expansion*>, <*join*>, <*replace*>, <*items*>)
Python Example	`oEditor.Expand("2mm", "ROUND", True, ["NAME:elements", "circle_0"])`

ExpandWithPorts (Layout Editor)

Scales an existing circle, square, or polygon and preserves ports. To expand an object without preserving ports, use Expand.

UI Access	**Draw** > **Expand**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*expansion*></td><td>String</td><td>Expansion amount, including unit. Value can be negative to shrink an object.</td></tr><tr><td><*join*></td><td>String</td><td>Corner style. One of: "ROUND" "MITER" "CORNER" </td></tr><tr><td><*items*></td><td>Array</td><td>Structured array containing object to expand (i.e., ["NAME:elements, "<name of object to expand>"]).</td></tr></table>
Return Value	None.

Python Syntax	ExpandWithPorts (<*expansion*>, <*join*>, <*items*>)
Python Example	`oEditor.ExpandWithPorts("2mm", "ROUND", ["NAME:elements", "circle_0"])`

FlipHorizontal (Layout Editor)

Use: Causes horizontal flipping of one or more specified objects (mirroring about a vertical axis given its X coordinate).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: **FlipHorizontal** <object_name_list>,//objects to be flipped <position_or_center> // (X of flip_axis / not used)

FlipVertical (Layout Editor)

Use: Causes vertical flipping of one or more specified objects (mirroring about a horizontal axis given its Y coordinate).

Syntax: **FlipVertical** <object_name_list>, //objects to be flipped <position_or_center> // not used, flip axis

Move (Layout Editor)

Use: Translate this Polygon by the vector specified in the provided Point. Return this Polygon.

Syntax: oPolygon.Move(<oPoint>)

Return Value: This Polygon object.

Paste (Layout Editor)

Use: Pastes the objects currently in clipboard; the argument specifies the offset of the new (copy) object(s) from their initial position(s).

Syntax: **Paste** Array("NAME:offset",

```
"xy:=",
Array(double, double)) // X/Y offset
```

Rotate (Layout Editor)

Use: Rotate this Polygon counter clockwise (CCW) about a center specified by the provided Point. Angle is in radians. Return this Polygon.

Syntax: oPolygon.Rotate(<AngRad>, <oPoint>)

Return Value: This Polygon object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: `oPolygon = oPolygon.Rotate(pi/2, oLayout.Point)`

Activation and Deactivation Methods

The following activation/deactivation methods are available.

Activate

DeactivateOpen

DeactivateShort

Delete

Activate (Layout Editor)

Activates one or more components.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Components>
<object_name></td></tr>
<tr><td><NAME:Components></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Components").</td></tr>
<tr><td><SelectedLayout></td><td>String</td><td>A comma-separated list of object names (e.g., "polygon_01", "polygon_02", "polygon_03", et cetera).</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Activate([<*NAME:Components*>, <*object_name*>])

Python Example	`oEditor.Activate(`
	`  [`
	`  "NAME:Components",`
	`  "polygon_1",`
	`  "polygon_2"`
	`  ])`

DeactivateOpen (Layout Editor)

Use: Causes deactivation of one or more components to *open circuit* state.

Syntax: **ActivateOpen** Array ("NAME:components",

<object_name>, // 1st component name

<object_name>, // 2nd component, if any

…) // etc

DeactivateShort (Layout Editor)

Use: Causes deactivation of one or more components to *short circuit* state.

Syntax: **ActivateShort** Array("NAME:components",

`<object_name>, // 1st component name`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<object_name>, // 2nd component, if any

…) // etc

Delete (Layout Editor)

Use: Deletes one or more objects.

Syntax: **Delete** <object_name_list>// names of the objects to be deleted

<object_name_list>:

Array ("*NAME:elements*", <object_name>, <object_name>, …)

Layout and Geometry Interrogation Methods

From the Layout scripting object, the following methods are available for layout and geometry interrogation.

The topics for this section include:

Layout Interrogation

Point Object

Polygon Object

Layout Interrogation (Layout Editor)

Methods employing Point and Polygon objects are specified in SI units (e.g. meters).

The following methods are available.

CreateLineFromPolygon (Layout Editor)

CreateObjectFromPolygon (Layout Editor)

FindObjects (Layout Editor

FilterObjectList (Layout Editor)

FindObjectsByPoint (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

FindObjectsByPolygon (Layout Editor)

GetBBox (Layout Editor)

GetCSObjects (Layout Editor)

GetPolygon (Layout Editor)

GetPolygonDef (Layout Editor)

Point (Layout Editor)

Polygon (Layout Editor)

CreateLineFromPolygon (Layout Editor)

Use: Create a line/trace object on the specified layer and net from the provided Polygon object. Net is optional.

Syntax: oLayout.CreateLineFromPolygon(<oPolygon>, <width>, <bendType>, <startCapType>, <endCapType>, <Layer>, <Net>)

Parameters: `width - line width, in meters`

bendType - {'corner', 'round'}

startCapType, endCapType - {'flat', 'extended', 'round'}

CreateObjectFromPolygon (Layout Editor)

Create a polygon on the specified layer and net.

UI Access	None.		
Parameters	Name	Type	Description
	<oPolygon>	Object	A polygon object used to define the new layout polygon.
	<Layer>	String	The name of the layer on which the new polygon will be created.
	<Net>	String	The net to assign to the new polygon. Optional.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	New polygon name.

Python Syntax	oLayout.CreateObjectFromPolygon(<oPolygon>, <Layer>, <Net>)

Python Example	`oLayout.CreateObjectFromPolygon(`
	`  oPolygon,`
	`  "Layer1",`
	`  "Net1"`
	`)`

FindObjects (Layout Editor)

Returns a list of object names using a key/value pair.

UI Access	N/A.

Parameters

Name	Type	Description
<key>	String	One of: • **"Name"** to search by object name. • **"Type"** to search by object type. Valid <value> strings for this type include: 'pin', 'via', 'rect', 'arc', 'line', 'poly', 'plg', 'circle void', 'line void', 'rect void', 'poly void', 'plg void', 'text', 'cell', 'Measurement', 'Port', 'Port Instance', 'Port Instance Port', 'Edge Port', 'component', 'CS', 'S3D', 'ViaGroup' • **"Layer"** to search by layer.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			An asterisk ('*') in the <value> string matches all layers. Vias and pins match using their defined layer range. 'Multi' matches (non via/pin) multi-layer objects. • **"Net"** to search by net assignment.
	<value>	String	Specific query. Queries use a basic string match. For instance, an asterisk ('*') matches anything. All matching is case insensitive.
Return Value	List of object names.		

Python Syntax	FindObjects (*<key>*, *<value>*)
Python Example	`oLayout.FindObjects('Type','Via')`

FilterObjectList (Layout Editor)

Filters a list of object names using a key/value pair. See: FindObjects.

UI Access	N/A.		
Parameters	Name	Type	Description
	<key>	String	One of: • **"Name"** to search by object name. • **"Type"** to search by object type. Valid <value> strings for this type include: 'pin', 'via', 'rect', 'arc', 'line', 'poly', 'plg', 'circle void', 'line void', 'rect void', 'poly void', 'plg void', 'text', 'cell', 'Measurement', 'Port', 'Port Instance', 'Port Instance Port', 'Edge

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Port', 'component', 'CS', 'S3D', 'ViaGroup' • **"Layer"** to search by layer. An asterisk ('*') in the <value> string matches all layers. Vias and pins match using their defined layer range. 'Multi' matches (non via/pin) multi-layer objects. • **"Net"** to search by net assignment.
	<*value*>	String	Specific query. Queries use a basic string match. For instance, an asterisk ('*') matches anything. All matching is case insensitive.
Return Value	Filtered list of object names.		

Python Syntax	FilterObjectList (<*key*>, <*value*>)
Python Example	`RFIN_pins = oLayout.FilterObjectList('Type', 'Pin', oLayout.FindObjects('Net','RFIN'))`

FindObjectsByPoint (Layout Editor)

Use: Get a list of all objects intersected by the specified Point object on a given layer.

Syntax: oLayout.FindObjectsByPoint(<oPoint>, <Layer>)

Return Value: Object list.

FindObjectsByPolygon (Layout Editor)

Use: Get a list of all objects intersecting the specified Polygon object on the given layer.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: oLayout.FindObjectsByPolygon(<oPolygopn>, <Layer>)

Return Value: Object list.

GetBBox (Layout Editor)

Use: Get a Polygon Object defining the bounding box for the specified object name, or None if the object does not exist.

Syntax: oLayout.GetBBox(<Name>)

Return Value: Polygon object.

GetCSObjects (Layout Editor)

Use: Given a coordinate system name, returns a list of the objects inside. Coordinate system names can be found using FindObjects ('Type', 'CS'). See Find Objects.

Syntax: oLayout.GetCSObjects(<CS_name>)

Return Value: List of objects in the coordinate system.

GetPolygon (Layout Editor)

Use: Get a Polygon Object for the specified object name, or None if the object does not exist. Returned polygon is as rendered, in SI units

Syntax: oLayout.GetPolygon(<Name>)

Return Value: Polygon object.

GetPolygonDef (Layout Editor)

Use: Get a Polygon Object for the specified object name, or None if the object does not exist. Returned polygon is as rendered, in SI units. For trace types, this corresponds to the center line.

Syntax: oLayout.GetPolygonDef(<Name>)

Return Value: Polygon object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Point (Layout Editor)

*Use:*Create and return a Point Object.

*Syntax:*oLayout.Point()

*Return Value:*Point object.

Polygon (Layout Editor)

Use: Create and return a Polygon Object.

Syntax: oLayout.Polygon()

Return Value: Polygon object.

Point Object (Layout Editor)

The Point object provides a point container in a 2D Cartesian coordinate system. Point objects have no link to specific Layout primitives. All values and interfaces are specified in meters.

A Layout object is used to create a Point as:

```
oPoint = oLayout.Point()
```

The following methods are available.

Set

SetX

GetX

SetY

GetY

SetArc

IsArc

IsEqual

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Mag

Distance

Cross

Move

Rotate

Normalize

DistanceFromLine

ClosestPointOnLine

Set (Layout Editor)

Use: Set both the x and y -coordinates of the Point, in meters. Return *this* Point.

Syntax: oPoint.Set(<x>, <y>)

Return Value: This Point object.

```
Example: oPoint = oPoint.Set(1e-3, 1e-3)
```

SetX (Layout Editor)

Use: Set the x-coordinate of the Point, in meters. Return *this* Point.

Syntax: oPoint.SetX(<x>)

Return Value: This Point object.

```
Example: oPoint = oPoint.SetX(1e-3)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetX (Layout Editor)

*Use:*Get the x-coordinate of the Point, in meters.

*Syntax:*oPoint.GetX()

*Return Value:*X-coordinate.

SetY (Layout Editor)

Use: Set the y-coordinate of the Point, in meters. Return *this* Point.

Syntax: oPoint.SetY(<y>)

Return Value: This Point object.

```
Example: oPoint = oPoint.SetY(1e-3)
```

GetY (Layout Editor)

Use: Get the y-coordinate of the Point, in meters.

Syntax: oPoint.GetY()

Return Value: Y-coordinate.

SetArc (Layout Editor)

*Use:*Set whether the given point is an arc. For arc-points, the x-coordinate is used to specify signed height. Return *this* Point.

*Syntax:*oPoint.SetArc(<isArc>)

*Return Value:*This Point object.

```
Example:oPoint = oPoint.SetArc(True)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IsArc (Layout Editor)

Use: Check whether this Polygon is defined by an arc-segment.

Syntax: oPolygon.IsArc()

IsEqual (Layout Editor)

Use: Test whether this point is equal to another. If tolerance is not specified, a default 1e-9 is used.

Syntax: oPoint.IsEqual(<oPoint>, [<Tolerance>])

Mag (Layout Editor)

*Use:*Get the magnitude of the given Point

*Syntax:*oPoint.Mag()

Distance (Layout Editor)

Use: Get the distance from this point to another.

Syntax: oPointThis.Distance(oPointOther)

Cross (Layout Editor)

Use: Get the cross product of this point with another. In 2D, this is simply the determinant of [x1 x2; y1 y2].

Syntax: oPointThis.Cross(oPointOther)

Move (Layout Editor)

Use: Translate this Polygon by the vector specified in the provided Point. Return this Polygon.

Syntax: oPolygon.Move(<oPoint>)

Return Value: This Polygon object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Rotate (Layout Editor)

Use: Rotate this Polygon counter clockwise (CCW) about a center specified by the provided Point. Angle is in radians. Return this Polygon.

Syntax: oPolygon.Rotate(<AngRad>, <oPoint>)

Return Value: This Polygon object.

Example: `oPolygon = oPolygon.Rotate(pi/2, oLayout.Point)`

Normalize (Layout Editor)

Use: Normalize this point. Return this point.

Syntax: oPoint.Normalize()

Return Value: This Point object.

DistanceFromLine (Layout Editor)

Use: Get the distance from this point to a line defined by two additional points.

Syntax: oPoint.DistanceFromLine(<oPoint>,<oPoint>)

ClosestPointOnLine (Layout Editor)

Returns the closest point on a line defined by two input points.

<table>
<tr><td>UI Access</td><td colspan="3">None</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><point1></td><td>Integer</td><td>First point defining the line.</td></tr>
<tr><td><point2></td><td>Integer</td><td>Second point defining the line.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">Returns the closest point on the line, as a point object.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ClosestPointOnLine(<*point1*>, <*point2*>)

Python Example	`oPoint.ClosestPointOnLine(`
	`  point1,`
	`  point2`
	`)`

Polygon Object (Layout Editor)

The Polygon object provides a polygon container in a 2D Cartesian coordinate system. Objects have no link to specific layout primitives. All values and interfaces are specified in meters.

A layout object can be used to create an empty, and open, Polygon as:

```
oPolygon = oLayout.Polygon()
```

The following methods are available.

AddPoint

SetClosed

IsClosed

Move

Rotate

Scale

MirrorX

GetPoints

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddHole

HasHoles

GetHoles

HasArcs

HasSelfIntersections

BBoxLL

BBoxUR

IsParametric

IsConvex

IsPoint

IsSegment

IsArc

IsBox

IsCircle

GetBoundingCircleCenter

GetBoundingCircleRadius

Area

PointInPolygon

CircleIntersectsPolygon

GetIntersectionType

GetClosestPoint

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetClosestPoints

Unite

Intersect

Subtract

Xor

AddPoint (Layout Editor)

Adds a point to a polygon.

UI Access	None
Parameters	Name | Type | Description <*polygon_name*> | String | Name of the selected polygon.
Return Value	The name of the selected polygon.

Python Syntax	AddPoint(<polygon_name>)
Python Example	`Polygon.AddPoint("polygon_name")`

SetClosed (Layout Editor)

Use: Set whether a polygon is open or closed. Return this Polygon.

Syntax: oPolygon.SetClosed(isClosed)

Return Value: This Polygon object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: `oPolygon = oPolygon.SetClosed(True)`

IsClosed (Layout Editor)

Use: Get whether a polygon is open or closed.

Syntax: oPolygon.IsClosed()

Move (Layout Editor)

Use: Translate this Polygon by the vector specified in the provided Point. Return this Polygon.

Syntax: oPolygon.Move(<oPoint>)

Return Value: This Polygon object.

Rotate (Layout Editor)

Use: Rotate this Polygon counter clockwise (CCW) about a center specified by the provided Point. Angle is in radians. Return this Polygon.

Syntax: oPolygon.Rotate(<AngRad>, <oPoint>)

Return Value: This Polygon object.

Example: `oPolygon = oPolygon.Rotate(pi/2, oLayout.Point)`

Scale (Layout Editor)

Scales the polygon by the specified factor.

UI Access	N/A.		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<factor>	Integer	Scaling factor.
	<centerPoint>	Object	Any oPoint object. See: Point Object.
Return Value	Polygon object.		

Python Syntax	Scale(*<factor>*, *<centerPoint>*)
Python Example	`oPolygon.Scale(2, oPoint)`

MirrorX (Layout Editor)

Use: Mirror this Polygon about the line defined by the provided X-coordinate. Return this Polygon.

Syntax: oPolygon.MirrorX(x-coord)

Return Value: This Polygon object.

GetPoints (Layout Editor)

Use: Get a list of Point objects defining this Polygon.

Syntax: oPolygon.GetPoints()

Return Value: List of Point objects.

AddHole (Layout Editor)

Adds a hole to a polygon.

UI Access	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*polygon_name*></td><td>String</td><td>Name of the selected polygon.</td></tr></table>
Return Value	The name of the selected polygon.

Python Syntax	AddHole(<polygon_name>)
Python Example	`Polygon.AddHole("polygon_name")`

HasHoles (Layout Editor)

Use: Get whether a Polygon has holes.

Syntax: oPolygon.HasHoles()

GetHoles (Layout Editor)

Use: Get a list of holes owned by this Polygon.

Syntax: oPolygon.GetHoles()

Return Value: List of holes as Polygon objects.

HasArcs (Layout Editor)

Use: Get whether this Polygon has any arc-segments.

Syntax: oPolygon.HasArcs()

HasSelfIntersections (Layout Editor)

Use: Check whether this Polygon has any self-intersections. Default tolerance is 1e-9, or it may be explicitly provided.

Syntax: oPolygon.HasSelfIntersections(<tolerance>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

BBoxLL (Layout Editor)

Returns the lower-left point of the bounding-box region for the specified polygon.

UI Access	None
Parameters	None
Return Value	A point object defining the lower-left coordinate.

Python Syntax	oPolygon.BBoxLL()
Python Example	`oPolygon.BBoxLL()`

BBoxUR (Layout Editor)

Returns the upper-right point of the bounding-box region for the specified polygon.

UI Access	None
Parameters	None
Return Value	A point object defining the upper-right coordinate.

Python Syntax	oPolygon.BBoxUR()
Python Example	`oPolygon.BBoxUR()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IsParametric (Layout Editor)

Use: Check whether this Polygon is defined parametrically.

Syntax: oPolygon.IsParametric()

IsConvex (Layout Editor)

Use: Check whether this Polygon is convex.

Syntax: oPolygon.IsConvex()

IsPoint (Layout Editor)

Use: Check whether this Polygon is defined by a point.

Syntax: oPolygon.IsPoint()

IsSegment (Layout Editor)

Use: Check whether this Polygon is defined by a segment.

Syntax: oPolygon.IsSegment

IsArc (Layout Editor)

Use: Check whether this Polygon is defined by an arc-segment.

Syntax: oPolygon.IsArc()

IsBox (Layout Editor)

Use: Check whether this Polygon defines a box.

Syntax: oPolygon.IsBox()

IsCircle (Layout Editor)

Use: Check whether this Polygon defines a circle.

Syntax: oPolygon.IsCircle()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetBoundingCircleCenter (Layout Editor)

Use: Get a Point defining the center of a bounding circle for this Polygon.

Syntax: oPolygon.GetBoundingCircleCenter()

Return Value: A Point object defining the bounding circle's center coordinate.

GetBoundingCircleRadius (Layout Editor)

Use: Get the radius of a bounding circle for this Polygon.

Syntax: oPolygon.GetBoundingCircleRadius()

Return Value: The bouding circle's radius.

Area (Layout Editor)

Returns the area of the polygon.

UI Access	None
Parameters	None
Return Value	Area of the polygon, in square meters.

Python Syntax	oPolygon.Area()
Python Example	`oPolygon.Area()`

PointInPolygon (Layout Editor)

Use: Test whether the provided Point is in this Polygon. Points on the Polygon's boundary are considered in the Polygon.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: oPolygon.PointInPolygon(<oPoint>)

CircleIntersectsPolygon (Layout Editor)

Tests whether the circle defined by the input point and radius intersects this polygon. Circles tangent to the polygon's boundary are considered intersecting.

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><oPointCenter></td><td>object</td><td>The center point of the circle.</td></tr><tr><td><Radius></td><td>float</td><td>The radius of the circle.</td></tr></table>
Return Value	Boolean (**True** if intersecting, **False** if not)

Python Syntax	CircleIntersectsPolygon(<oPointCenter>, <Radius>)

Python Example	`oPolygon.CircleIntersectsPolygon(`
	`  "Point1",`
	`  5.0`
	`)`

GetIntersectionType (Layout Editor)

Use: Get the intersection type of this Polygon with another. A tolerance can be optionally specified, or the default 1e-9 will be used.

Syntax: oPolygon.GetIntersectionType(<oPolygon>)

Return Value: Integer:

0 - No intersection

1 - This fully inside other

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

2 - Other fully inside this

3 - Common contour points

4 - Undefined intersection

typ2 = oPolygon.GetIntersectionType(oPolygonOther, 1e-6)

GetClosestPoint (Layout Editor)

Use: Find the closest Point on this Polygon's boundary to the Point provided.

Syntax: oPolygon.GetClosestPoint(<oPoint>)

Return Value: A Point object defining the closest location on this Polygon’s boundary.

GetClosestPoints (Layout Editor)

Use: Given another Polygon, find the closest Points on each. The closest Point on this Polygon is returned as the first Point in the list. The closest Point on the other Polygon is returned as the last (or second) Point in the list.

Syntax: oPolygon.GetClosestPoints(oPolygon)

Return Value: A list of Point objects.

Unite (Layout Editor)

Use: Causes Boolean uniting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: **Unite** Array ("NAME: *primitives*",

```
<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
…) // etc
Example:
oEditor.Unite ["NAME:primitives", "circle_0", "rect_2"]
```

Intersect (Layout Editor)

Use: Causes Boolean intersecting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: **Intersect** Array ("NAME: *primitives*",

```
<object_name>, // 1st primitive name
<object_name>, // 2nd primitive, if any
        …) // etc
```

Subtract (Layout Editor)

Use: Causes boolean subtracting of one or more *primitive* (polygons, rectangles, lines, or circles) object(s) from another one.

Syntax: **Subtract** Array ("NAME: *primitives*",

```
<object_name>, // Primitive to subtract from
<object_name>, // 1nd primitive to subtract, if any
…) // etc
Example:
oEditor.Intersect Subtract ("NAME:primitives", "circle_0", "rect_2")
```

Xor (Layout Editor)

Use: Exclusive or (XOR) this Polygon with another and return as a list of Polygons.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: oPolygon.Xor(oPolygon)

Return Value: Xor as a list of Polygon objects.

Example:

```
resList = oPolygon.Xor(oPolygonOther)
```

Boolean Operations on Primitives

The following boolean operations on primitives are available.

ConvertPrimitives (Layout Editor)

ConvertPrimitivestoVias (Layout Editor)

Unite (Layout Editor)

Intersect (Layout Editor)

SplitRegion (Layout Editor)

Subtract (Layout Editor)

ToggleViaPin (Symbol Editor)

ConvertPrimitives (Layout Editor)

Converts the selected primitives into 2.5D via structures.

UI Access	None
Parameters	Name: *<primvia_data>*; Type: Array; Description: List of primitive objects to convert. Each entry should reference a valid primitive name.
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ConvertPrimitives(<*primvia_data*>)

Python Example	`oEditor.ConvertPrimitives(`
	`  ["primitive_01",`
	`  "primitive_02"`
	`  ])`

ConvertPrimitivestoVias (Layout Editor)

Converts primitives in the active design to vias.

UI Access	After selecting one or more primitives, navigate to **Draw** > **Convert** > **Primitives to Vias**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:elements> <TargetPrimitives> </td></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "elements").</td></tr><tr><td><*TargetPrimitives*></td><td>String</td><td>Names of the target primitives to convert to vias, separated by commas.</td></tr></table>
Return Value	None.

Python Syntax	ConvertPrimitivestoVias([<NAME:elements>, <TargetPrimitives>])
Python Example	`oEditor.ConvertPrimitivestoVias(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:elements", "rect_395"])

Unite (Layout Editor)

Use: Causes Boolean uniting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: **Unite** Array ("NAME: *primitives*",

<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any

…) // etc

Example:

```
oEditor.Unite ["NAME:primitives", "circle_0", "rect_2"]
```

Intersect (Layout Editor)

Use: Causes Boolean intersecting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: **Intersect** Array ("NAME: *primitives*",

<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

…) // etc

SplitRegion (Layout Editor)

Split geometry against a polygon region.

UI Access	After the polygon is created and the design and polygon are selected, right-click and select **Draw** > **Split Polygon Region**.
Parameters	None.
Return Value	`"Name:Poly_#"`

Python Syntax	SplitRegion

Python Example	`oEditor.SplitRegion(`
	`  [`
	`  "NAME:ExampleName",`
	`  "Poly_1"`
	`  ])`

Subtract (Layout Editor)

Use: Causes boolean subtracting of one or more *primitive* (polygons, rectangles, lines, or circles) object(s) from another one.

Syntax: **Subtract** Array ("NAME: *primitives*",

`<object_name>, // Primitive to subtract from`

`<object_name>, // 1nd primitive to subtract, if any`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
…) // etc
```

Example:

```
oEditor.Intersect Subtract ("NAME:primitives", "circle_0", "rect_2")
```

ToggleViaPin (Symbol Editor)

Selected pins are changed to vias. Selected vias are changed to pins.

Syntax: ToggleViaPin <NAME:elements", <object_name> ... // objects to be toggled

Return Value: None.

Parameters: <object_name>

Type: <String>

Example: oEditor.ToggleViaPin ["NAME:elements", "via_195"]

Coordinate System Methods

The following scripts are availabe for use with the Designer Coordinate System.

CreateCS (Layout Editor)

ClearRelative (Layout Editor)

Create3DStructure (Layout Editor)

GetCSObjects (Layout Editor)

Group (Layout Editor)

CreateGroupSelected (Layout Editor)

PositionRelative (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetCS (Layout Editor)

Ungroup (Layout Editor)

CreateCS (Layout Editor)

Inserts a new coordinate system (CS) into the currently active Layout design.

<table>
<tr><td>UI Access</td><td>From Draw, select Coordinate System > Create.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Required header string identifying the parameter block.</td></tr>
<tr><td><Full_Definition></td><td>Array</td><td>Contains:
<code>Type</code>: Type of the coordinate system (e.g., "CS")
<code>N</code>: Name of the coordinate system</td></tr>
<tr><td><Type></td><td>String</td><td>Type of the coordinate system.</td></tr>
<tr><td><N></td><td>String</td><td>Name of the coordinate system.</td></tr>
<tr><td><Placement></td><td>Array</td><td>Contains:
<code>X</code>: X origin coordinate
<code>Y</code>: Y origin coordinate</td></tr>
<tr><td><X></td><td>Double</td><td>X origin coordinate.</td></tr>
<tr><td><Y></td><td>Double</td><td>Y origin coordinate.</td></tr>
<tr><td><Clf></td><td>Boolean</td><td>True if a color is being specified by the next parameter.</td></tr>
<tr><td><Col></td><td>Integer</td><td>Color index for the coordinate system. Only used if <code>Clf</code> is true.</td></tr>
<tr><td><Name></td><td>String</td><td>Requested name for the coordinate system.</td></tr>
<tr><td><RelPos></td><td>Array</td><td>Contains:
<code>t</code>: Edge type ("pe" for primitive, "pse" for via)
<code>from</code>: Primitive or via name
<code>pos</code>: Position along the edge (0–1)
<code>et</code>: Edge type ("pe" or "pse")</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Name	Type	Description
			• `prim`: Primitive name (if using primitive edge) • `edge`: Edge number (if using primitive edge) • `sel`: Via name (if using via edge) • `layer`: Layer ID (if using via edge) • `sx`, `sy`: Start X, Y of arc (via edge) • `ex`, `ey`: End X, Y of arc (via edge) • `h`: Arc height • `rad`: Arc size in radians
	<t>	String	Edge type: "pe" for primitive, "pse" for via.
	<from>	String	Name of the primitive or via.
	<pos>	Double	Position along the edge (0 = start, 1 = end).
	<et>	String	Edge type repeated: "pe" or "pse".
	<prim>	String	Primitive name (used for primitive edge).
	<edge>	Integer	Edge number on the primitive.
	<sel>	String	Via name (used for via edge).
	<layer>	Integer	Layer ID of the via pad.
	<sx>, *<sy>*	Double	X, Y location of the start point of the arc.
	<ex>, *<ey>*	Double	X, Y location of the end point of the arc.
	<h>	Double	Height of the arc (0 for straight edge).
	<rad>	Double	Arc size in radians (0 for straight edge).
Return Value	Returns the name of the created relative coordinate system.		

Python Syntax	CreateCS([<NAME:Contents>, <Full_Definition>, <Placement>, <Clf>, <Col>, <Name>, <RelPos>])
Python Example	`oEditor.CreateCS([`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "NAME:Contents",
  "Full_Definition:=", ["Type:=", "CS", "N:=", "MyCS"],
  "Placement:=", ["X:=", 0.0, "Y:=", 0.0],
  "Clf:=", True,
  "Col:=", 3,
  "Name:=", "MyCS",
  "RelPos:=", [
  "t:=", "pe",
  "from:=", "Prim1",
  "pos:=", 0.5,
  "et:=", "pe",
  "prim:=", "Prim1",
  "edge:=", 2
  ]
])
```

ClearRelative (Layout Editor)

Clears the relative positioning between a coordinate system (CS) and another object.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:elements>	String	Required first parameter which must have the specified value (i.e., "ele-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ments").
	<*CS_name*>	String	Name of the coordinate system whose relative positioning is to be cleared.
Return Value	None		

Python Syntax	ClearRelative(<*NAME:elements*>, <*CS_name*>)

Python Example	`oEditor.ClearRelative(`
	`  ["NAME:Elements",`
	`  "CS1",`
	`  "CS2"`
	`  ])`

Create3DStructure (Layout Editor)

Place a set of primitives into a 3D structure coordinate system (i.e., a 3D via or cross-layer plate).

UI Access	None		
Parameters	Name	Type	Description
	<*3DStructureName*>	String	Name of the 3D structure.
	<*NAME:Elements*>	String	Object descriptor header.
	<*ElementN*	String	Names of layout primitives to include in the structure, separated by commas.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Create3DStructure(<3DStructureName>, [<NAME:Elements>, <Element1>, <Element2>, …])

Python Example	`oEditor.Create3DStructure(`
	`  "ViaStructure", [`
	`  "NAME:Elements",`
	`  "ViaTop",`
	`  "ViaBarrel",`
	`  "ViaBottom"`
	`])`

GetCSObjects (Layout Editor)

Use: Given a coordinate system name, returns a list of the objects inside. Coordinate system names can be found using FindObjects ('Type', 'CS'). See Find Objects.

Syntax: oLayout.GetCSObjects(<CS_name>)

Return Value: List of objects in the coordinate system.

Group (Layout Editor)

Use: Groups a set of primitives into a coordinate system.

Command: None.

Syntax: oEditor.Group <CS name>, Array("NAME:elements", <element1>, <element2>, …)

Return Value: None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateGroupSelected (Layout Editor)

Groups a set of layout primitives into a subdesign.

UI Access	From **Layout Editor**, navigate to to **Layout** > **Group into Subdesign**.		
Parameters	Name	Type	Description
	<NAME:elements>	String	Required header string identifying the parameter block.
	<Selectable Name>	String	Name of each primitive to include in the group. Multiple entries allowed.
Return Value	None		

Python Syntax	CreateGroupSelected([*elements*])

Python Example	`oEditor.CreateGroupSelected([`
	`  "NAME:elements",`
	`  "Selectable Name:=", "Resistor1",`
	`  "Selectable Name:=", "Capacitor2"`
	`])`

PositionRelative (Layout Editor)

Use: Position a coordinate system (CS), including 3D structures and components, relative to another object.

Command: Draw > Position Relative

Syntax: There are two forms available:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PositionRelative

Array("NAME:Contents",

"RelPos:=", Array(<edge description>)),

Array("NAME:elements", <coordinate system name>, < coordinate system name >, …)

PositionRelative Array("NAME:Contents",

"RelPos:=", Array("t:=", "pr", "from:=", <edge port or pin name>)),

Array("NAME:elements", <coordinate system name>, <coordinate system name>, …)

Return Value: None.

Parameters:

Array("NAME:elements", "CS_23")

<edge description> for primitive edges

"t:=", "pe", "from:=", <"prim">, "pos:=", <edge position>, "et:=", "pe", "prim:=", <"prim">, "edge:=", <edge#>

<"prim">: text that is the primitive name

<edge position>

double between 0 and 1, inclusive

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

0 indicates the start of the edge

1 indicates the end of the edge

values in between are positions along the edge

<edge#>: integer that is the edge number on the primitive

<edge description> for via edges

"t:=", "pe", "from:=", <"via">, "pos:=", <edge position>, "et:=", "pse",
"sel:=", <"via">, "layer:=", <layer id>, "sx:=", <start X location>,
"sy:=", <start Y location>, "ex:=", <end X location>, "ey:=",
<end Y location>, "h:=", <arc height>, "rad:=", <radians>

<"via">: text that is the name of the via to use

<layer id>: an integer that is the id of the layer of the pad of the via to use

<start X location>, <start Y Location>:
doubles that are the X, Y location of the start point of the edge arc

<end X location>, <end Y Location>:
doubles that are the X, Y location of the end point of the edge arc

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<arc height>: double giving the height of the edge arc (0 for a straight edge)

<radians>: double giving the arc size in radians (0 for a straight edge)

< coordinate system name >

Text that contains the name of the coordinate system

<edge port or pin name>

Text that contains the name of the edge port or via

SetCS (Layout Editor)

Use: Activate a coordinate system (CS).

Command: None.

Syntax: oEditor.SetCS <CS name or blank>

Return Value: None.

Example:
```
oEditor.SetCS "CS_7" // activate CS_7
oEditor.SetCS "" // activate the 'global CS', i.e. no CS is active
```

Ungroup (Layout Editor)

Use: Reverses a group operation; deletes a coordinate system but leaves the primitives.

Command: None.

Syntax: oEditor.Ungroup Array("NAME:elements", <CS1>, <CS2>, …)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None.

```
Example: oEditor.Ungroup ["NAME:elements", "CS_7"]
```

IC Editor Script Commands (IC Mode Only)

IC Editor scripting commands are accessible once **Electronics Desktop** has been placed in IC Editor mode (i.e., **IC** mode).

Enter the following commands to place **Electronics Desktop** in **IC** mode and access the *IC Editor* interface by executing the **SetActiveEditor** command.

```
oDesign = oProject.SetActiveDesign("Design_Example1")
oDesign.SetDesignMode("IC")
oEditor = oDesign.SetActiveEditor("ICEditor")
```

The call to each scripting method contains three elements:

- Layout editor
- Method name
- Method argument(s) (if any)

For example, in Visual Basic:

```
oEditor.Delete ("rect_1")
```

The behavior of the scripting methods is similar to the corresponding commands in the **Layout Editor**. This correspondence is evident when recording a script in Layout Editor (i.e., **Tools** > **Record Script**).

The following commands can only be executed from the **IC** mode environment.

ApplyTechnology (IC Editor)

ChangeActiveLayout (IC Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CompressPrimitives (IC Editor)

CreateViaGroupsOnLayer (IC Editor)

ReconstructArcs (IC Editor)

ReconstructArcsOnLayer (IC Editor)

RemoveSmall Holes (IC Editor)

RemoveSmallLines (IC Editor)

RemoveSmallMetalIslands (IC Editor)

ResetTechnology (IC Editor)

SnapPrimitives (IC Editor)

SnapPrimitivesOnLayer (IC Editor)

SnapVias (IC Editor)

SnapViasOnLayer (IC Editor)

UnitePrimitivesOnLayer (IC Editor)

WrapGeometries (IC Editor)

WrapGeometry (IC Editor)

WrapGeometry (IC Editor)

Common to General Mode

The following commands are common to **General** mode and **IC** mode environment. Unless otherwise noted, the commands function identically in either environment (e.g., **RemoveLayer** is divided into two sections for each mode since the command function is similar but not identical).

AddObject (IC Editor and Layout Editor)

AddPinGroupRefPort (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddPortsToNet (Layout Editor)

AddRefPort (Layout Editor)

AssignRefPort (Layout Editor)

ChangeLayer (Layout Editor)

ChangeLayers (Layout Editor)

ChangeOptions (Layout Editor)

ChangeProperty (Schematic Editor and Layout Editor)

ClearRefPort (Layout Editor)

ClipPlane (Layout Editor)

ConvertPrimitives (Layout Editor)

ConvertPrimitivestoVias (Layout Editor)

Copy (Layout Editor)

CreateCircle (Layout Editor)

CreateCircuitPort (Layout Editor)

CreateComponent (Layout Editor)

CreateDifferentialPair (Layout Editor)

CreateEdgePort (Layout Editor)

CreateLine (Layout Editor)

CreateLineFromPolygon (Layout Editor)

CreateNetClass (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateObjectFromPolygon (Layout Editor)
CreatePinGroup (Layout Editor)
CreatePinGroupPort (Layout Editor)
CreatePinGroups (Layout Editor)
CreatePolygon (Layout Editor)
CreatePortsOnComponents (Layout Editor)
CreatePortsOnComponentsByNet (Layout Editor)
CreateRectangle (Layout Editor)
CreateViaGroups (Layout Editor)
CutOutSubDesign (Layout Editor)
Delete (Layout Editor)
DeleteNets (Layout Editor)
DeletePinGroup (Layout Editor)
DelNetClass (Layout Editor)
DissolveComponents (Layout Editor)
Duplicate (Layout Editor)
DuplicateAcrossLyrs (Layout Editor)
Edit (Layout Editor)
EditComponent (Layout Editor)
EditPinGroup (Layout Editor)
EnableComponents (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportStackupXML (Layout Editor)
FilterObjectList (Layout Editor)
FindObjects (Layout Editor)
FindObjectsByPoint (Layout Editor)
FindObjectsByPolygon (Layout Editor)
GetActiveUnits (Layout Editor)
GetAllLayerNames (Layout Editor)
GetBBox (Layout Editor)
GetCSObjects (Layout Editor)
GetEditorName (Layout Editor)
GetLayerInfo (Layout Editor)
GetMaterialList (Layout Editor)
GetNetClasses (Layout Editor)
GetNetClassNets (Layout Editor)
GetNets (Layout Editor)
GetPolygon (Layout Editor)
GetPolygonDef (Layout Editor)
GetPolygonVoids (Layout Editor)
GetPortInfo (Layout Editor)
GetProperties (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropertyValue (Layout Editor)

GetSelections (Layout Editor)

GetStackupLayerNames (Layout Editor)

HighlightNet (Layout Editor)

ImportStackupXML (Layout Editor)

Intersect (Layout Editor)

ModifyDifferentialPair (Layout Editor)

ModifyNetClass (Layout Editor)

Move (Layout Editor)

Paste (Layout Editor)

Point (Layout Editor)

Polygon (Layout Editor)

RemoveLayer (Layout Editor)

RemovePort (Symbol Editor)

RemovePortsFromAllNets (Layout Editor)

RemovePortsOnComponents (Layout Editor)

ResetDifferentialPairs (Layout Editor)

Select (Layout Editor)

SelectAll (Layout Editor)

SelectNetConnected (Layout Editor)

SelectPhysicallyConnected (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetActiveUnits (Layout Editor)

SetCS (Layout Editor)

SetHfssExtentsVisible (Layout Editor)

SetHfssPortsVisible (Layout Editor)

SetNetColor (Layout Editor)

SetNetVisible (Layout Editor)

SetPropertyValue (Layout Editor)

Subtract (Layout Editor)

ToggleNetHighlight (Layout Editor)

ToggleViaPin (Symbol Editor)

Ungroup (Layout Editor)

Unite (Layout Editor)

UnselectAll (Layout Editor)

ApplyTechnology (IC Editor)

Applies the specified technology to the active design's stackup.

<table>
<tr><td>UI Access</td><td>From the Layout ribbon, select Apply Technology from the Technology drop-down menu.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><DefName></td><td>String</td><td>Name of technology to apply to the active design's stackup.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ApplyTechnology(<DefName>)
Python Example	`oEditor.ApplyTechnology("Model.vlc.tech")`

ChangeActiveLayout (IC Editor)

Switches the active layout.

UI Access	Navigate to the **Cells** window > **Hierarchical** tab. Then double-click the appropriate instance in the hierarchical tree.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:CellInstances> <SelectedLayout> </td></tr><tr><td><*NAME:CellInstances*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "CellInstances").</td></tr><tr><td><*SelectedLayout*></td><td>Integer</td><td>A comma-separated list of two integers, where the first integer always represents the top of the hierarchical tree (i.e., "0") and the second identifies the layer to make active (i.e., "1", "2", "3", et cetera).</td></tr></table>
Return Value	None

Python Syntax	ChangeActiveLayout([<*NAME:CellInstances*>, <*SelectedLayout*>])
Python Example	`oEditor.ChangeActiveLayout(` `  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NAME:CellInstances", "0", "1"])

Compress Primitives (IC Editor)

Compress geometrically identical primitives to improve system and simulation performance.

UI Access	From the **Project Manager** window, right-click the active design and select **Compress Primitives**.
Parameters	None
Return Value	None

Python Syntax	CompressPrimitives()
Python Example	`oEditor.CompressPrimitives()`

CreateMeshRegion (IC Editor)

Designate a mesh region in a design and create partitions for simulation.

UI Access	From the **Layout Editor**, select a component. Then right-click and select **Mesh Region** > **Create**.
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<MeshRegionSettings>	Array	Structured array containing these parameters in order: • <NAME:MeshRegionSettings> • <NumPrimaryPartitions> • <NumSecondaryPartitions> • <XY> • <PosZ> • <NegZ> • <UseActiveNets> • <InclRef> • <ExtentType>
<NAME:MeshRegionSettings>	String	Identifies the contents of the array (i.e., "MeshRegionSettings").
<NumPrimaryPartitions>	Integer	Number of primary partitions to create in the new mesh region.
<NumSecondaryPartitions>	Integer	Number of secondary partitions to create in the new mesh region.
<XY>	Integer	Horizontal padding on both sides of the new mesh.
<PosZ>	Integer	Vertical padding above the new mesh region.
<NegZ>	Integer	Vertical padding below the new mesh region.
<UseActiveNets>	Boolean	**True** will create a new mesh region defined by the active nets in the design. **False** will create a new mesh region defined by the dielectric extents in the design.
<InclRef>	Boolean	**True** will include bot positive nets and reference nets in the definition of the new mesh region, if <UseActiveNets> is **True**. **False** will not include bot positive nets and reference nets.
<ExtentType>	Integer	Define the new mesh region within a:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			0 — bounding box. **1** — conformal coat. **2** — convex hull.
Return Value	None.		

Python Syntax	CreatMeshRegion([<NAME:MeshRegionSettings>, <NumPrimaryPartitions>, <NumSecondaryPartitions>, <XY>, <PosZ>, <NegZ>, <UseActiveNets>, <InclRef>, <ExtentType>])

Python Example	`oEditor.CreateMeshRegion(`
	`[`
	`  "NAME:MeshRegionSettings",`
	`  "NumPrimaryPartitions:=", 6,`
	`  "NumSecondaryPartitions:=", 1,`
	`  "XY:=", "0.005",`
	`  "PosZ:=", "0.1",`
	`  "NegZ:=", "0.1",`
	`  "UseActiveNets:=", True,`
	`  "InclRef:=", False,`
	`  "ExtentType:=", 0`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateMeshRegionPartitionLine (IC Editor)

Create an additional partition line for a mesh region (i.e., CreateMeshRegion).

<table>
<tr><td>UI Access</td><td colspan="3">From the Layout Editor, select a mesh region. Then right-click and select Mesh Region > Add a partition line.</td></tr>
<tr><td rowspan="6">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Contents></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Contents>
<owner>
<mesh region partition lineGeometry>
<snap start>
<snap end></td></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Identifies the contents of the array (i.e., "Contents").</td></tr>
<tr><td><owner></td><td>String</td><td>Name of the mesh region in which to create the new partition line.</td></tr>
<tr><td><mesh region partition lineGeometry></td><td>Array</td><td>Name of the array containing the geometry of the new partition line, containing these parameters in order:
<Name>
<LayerName>
<lw>
<endstyle>
<StartCap>
<EndCap>
<n></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <U> • <x> • <y> • <MR>
<Name>	String	Name of the new partition line.
<LayerName>	Integer	Name of the layer on which to create the new partition line (i.e., "MeshRegions").
<lw>	Integer	Line width of the new partition line, including unit of measurement (i.e., "0.01mm").
<endstyle>	Integer	Of the new partition line, all vertices (i.e., points) will be: **0** — round. **1** — corner (i.e., square). **2** — mitered (i.e., extended to a point where it meets a corner).
<StartCap>	Integer	Of the new partition line, the starting point of the line will be: **0** — round. **1** — flat. **2** — extended (i.e., extended from the starting point beyond the actual beginning of its path).
<Endcap>	Integer	Of the new partition line, the ending point of the line will be: **0** — round. **1** — flat. **2** — extended (i.e., extended beyond the actual end point). **3** — Same as Start.
<n>	Integer	The number of vertices in the new partition line.
<U>	String	The unit of measurement by which the new partition will be created.
<x>	Integer	X coordinate of a point in the new partition line.
<y>	Integer	Y coordinate of a point in the new partition line.
<MR>	String	A number and unit of measurement representing the miter

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ratio (i.e., of the new partition line, assuming multiple segments, the size of the point where two segments meet).
	<*snap start*>	Boolean	**True** start of the new partition line will snap to a border of the mesh region.
	<*snap end*>	Boolean	**True** end of the new partition line will snap to a border of the mesh region.
Return Value	Returns the name of the new partition line.		

Python Syntax	CreatMeshRegionPartitionLine([<NAME:Contents>, <owner>, <mesh region partition lineGeometry>, [<Name>, <LayerName>, <lw>, <endstyle>, <StartCap>, <EndCap>, <n>, <U>, <x>, <y>, <MR>], <snap start>, <snap end>])

Python Example	`oEditor.CreateMeshRegionPartitionLine(`
	`[`
	`"NAME:Contents",`
	`"owner:=" , "meshregion__11",`
	`"mesh region partition lineGeometry:=",`
	`[ "Name:=" , "line__18",`
	`"LayerName:=" , "MeshRegions",`
	`"lw:=" , "0.01mm",`
	`"endstyle:=" , 0,`
	`"StartCap:=" , 0,`
	`"EndCap:=" , 0,`
	`"n:=" , 3,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "U:=" , "mm",
        "x:=" , 19,
        "y:=" , 9,
        "x:=" , 15,
        "y:=" , 4,
        "x:=" , 20,
        "y:=" , 1,
        "MR:=" , "600mm"],
    "snap start:=" , True,
    "snap end:=" , True
])
```

CreateViaGroupsOnLayer (IC Editor)

Creates a via group on the target layer.

UI Access	From the **Layers** window, right-click the target layer and select **Group Vias** > **Persistent** or **Non-Persistent**.
Parameters	See table below

Name	Type	Description
<*NAME*>	String	Name of the target layer.
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:Setting Contents> • <Group_By_Proximity>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Via_Grouping_Tolerance> • <Check_Containment> • <Via_Persistence>
	<NAME:Setting Contents>	String	Required first parameter which must have the specified value (i.e., "Setting Contents").
	<Group_By_Proximity>	Boolean	**True** groups via by proximity. **False** groups by range.
	<Via_Grouping_Tolerance>	Integer	When *<Group_By_Proximity>* = **False**, vias at or within this tolerance are grouped.
	<Check_Containment>	Boolean	**True** analyzes the placement of via primitives. When the possibility of an electrical short is detected, *<Check_Containment>* attempts to prevent the short. **False** deactivates the check.
	<Via_Persistence>	Boolean	**True** maintains the new via group. **False** selects the group of vias but does not maintain a new via group.
Return Value	None.		

Python Syntax	CreateViaGroupsOnLayer(<NAME>, [<NAME:Setting Contents>, <Group_By_Proximity>, <Via_Grouping_Tolerance>, <Check_Containment>, <Via_Persistence>])

Python Example	`oEditor.CreateViaGroupsOnLayer(`
	`["Trace",`
	`  "NAME:Setting Contents",`
	`  "group_by_proximity:=", "True",`
	`  "via_grouping_tolerance:=", "100um",`
	`  "check_containment:=", "True",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"via_persistence:=", "False"])

ReconstructArcs (IC Editor)

Fits arcs and circles to the selected polygons.

UI Access	After choosing polygons, select **Reconstruct Arcs** from the **Layout** ribbon.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:polygons> <Objects> <Arcs_Tolerance> </td></tr><tr><td><*NAME:polygons*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "polygons").</td></tr><tr><td><*Objects*></td><td>List</td><td>A comma-separated list of integers, where each integer represents a selected polygon (i.e., "poly_30", "poly_31" , "poly_32", etcetera)</td></tr><tr><td><*Arcs_Tolerance*></td><td>Integer</td><td>Arcs and circles at or within this tolerance are fitted.</td></tr></table>
Return Value	None.

Python Syntax	ReconstructArcs([<NAME:polygons>, <Objects>], <Arcs_Tolerance>)
Python Example	`oEditor.ReconstructArcs(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:polygons", "poly_30", "poly_31", "poly_32"], 0.00254)

ReconstructArcsOnLayer (IC Editor)

Fits arcs and circles to the selected polygons on a target layer.

UI Access	After choosing polygons, navigate to the **Layers** window. Then right-click the target layer and select **Reconstruct Arcs**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME*></td><td>String</td><td>Name of the target layer.</td></tr><tr><td><*Arcs_Tolerance*></td><td>Integer</td><td>Arcs and circles at or within this tolerance are fitted.</td></tr></table>
Return Value	None.

Python Syntax	ReconstructArcsOnLayer(<NAME>, <Arcs_Tolerance>)
Python Example	oEditor.ReconstructArcsOnLayer("Trace", 0.00254)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveSmallHoles (IC Editor)

Removes small holes at or within tolerance from the target layer.

<table>
<tr><td>UI Access</td><td>Select Remove Holes from the Layout ribbon.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Max_Area_Tolerance></td><td>Integer</td><td>Holes at or within this tolerance are removed.</td></tr>
<tr><td><Exclusion_Tolerance></td><td>Integer</td><td>Distance around nets to exclude from removal.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Nets>
<SelectedLayout></td></tr>
<tr><td><NAME:Nets></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Nets").</td></tr>
<tr><td><NetName></td><td>String</td><td>Name of the target net.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	AddObject(<ObjectType>, <LayerName>)

Python Example	`oEditor.RemoveSmallHoles("1000um2", "1um",`
	`  [`
	`  "NAME:Nets",`
	`  "net1"`
	`  ])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveSmallLines (IC Editor)

Removes small lines at or within tolerance from the target layer.

UI Access	From the **Layers** window, right-click the target layer and select **Remove Small Lines**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME*></td><td>String</td><td>Name of the target layer.</td></tr><tr><td><*Max_Length_Tolerance*></td><td>Integer</td><td>Lines at or within this tolerance are removed.</td></tr></table>
Return Value	None

Python Syntax	RemoveSmallLines(<NAME>, <Max_Length_Tolerance>)
Python Example	`oEditor.RemoveSmallLines("Trace", "1mm")`

RemoveSmallMetalIslands (IC Editor)

Removes primitives and vias at or within tolerance from the target layer.

UI Access	After choosing primitives, select **Remove Islands** from the **Layout Ribbon**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME*></td><td>String</td><td>Name of the target layer.</td></tr><tr><td><*Max_Area_Tolerance*></td><td>Integer</td><td>Primitives and vias at or within this tolerance are removed.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	RemoveSmallMetalIslands(<NAME>, <Max_Area_Tolerance>)
Python Example	`oEditor.RemoveSmallMetalIslands("Trace", "1mm")`

ResetTechnology (IC Editor)

Disassociates the applied technology from the active design's stackup.

UI Access	From the **Layout** ribbon, select **Reset Technology** from the **Technology** drop-down menu.
Parameters	None
Return Value	None

Python Syntax	ResetTechnology()
Python Example	`oEditor.ResetTechnology()`

SnapPrimitives (IC Editor)

Aligns the edge of the selected primitives.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>After choosing primitives, select Snap Primitives from the Layout ribbon.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><LAYER_NAME></td><td>String</td><td>Name of the target layer.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:primitives>
<Objects>
<Snap_Dis_Tolerance>
<CheckConnectivity></td></tr>
<tr><td><NAME:primitives></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "primitives").</td></tr>
<tr><td><Objects></td><td>List</td><td>A comma-separated list of integers, where each integer represents a selected primitive (i.e., "rect_0", "rect_1" , "rect_2", etcetera)</td></tr>
<tr><td><Snap_Dis_Tolerance></td><td>Integer</td><td>Primitives at or within this tolerance are snapped.</td></tr>
<tr><td><CheckConnectivity></td><td>Boolean</td><td>True analyzes the placement of primitives to ensure snapping does not violate the connectivity of the original objects.
False deactivates the check.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	SnapPrimitives(<LAYER_NAME> [<NAME:primitives>, <Objects>], <Snap_Dis_Tolerance>, <Check-Connectivity>)

Python Example	`oEditor.SnapPrimitives(`
	`["Trace",`
	`  "NAME:primitives",`
	`  "rect_0:=",`
	`  "rect_1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"rect_2",], "5mm", True)

SnapPrimitivesOnLayer (IC Editor)

Aligns the edge of the selected primitives on the target layer.

UI Access	From the **Layers** window, right-click the target layer and select **Snap Primitives**.
Parameters	**Name** / **Type** / **Description** *<LAYER_NAME>* / String / Name of the target layer. *<Snap_Dis_Tolerance>* / Integer / Primitives at or within this tolerance are snapped. *<CheckConnectivity>* / Boolean / **True** analyzes the placement of primitives to ensure snapping does not violate the connectivity of the original objects. **False** deactivates the check.
Return Value	None.

Python Syntax	SnapPrimitivesOnLayer(<LAYER_NAME>, <Snap_Dis_Tolerance>, <CheckConnectivity>)
Python Example	`oEditor.SnapPrimitivesOnLayer("Trace", "5mm", True)`

SnapVias (IC Editor)

Forces the edges of selected vias to snap to alignment.

UI Access	After choosing vias, select **Snap Vias** from the **Layout** ribbon.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:vias>
<Objects></td></tr>
<tr><td><NAME:vias></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "vias").</td></tr>
<tr><td><Objects></td><td>String</td><td>A comma-separated list of selected vias (i.e., "via_0", "via_1" , "via_2", etcetera).</td></tr>
<tr><td><Parameters2></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Setting Contents>
<IsFactorBased>
<SnapTol>
<AlignTol>
<RemoveDanglingVias></td></tr>
<tr><td><NAME:Setting Contents></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Setting Contents").</td></tr>
<tr><td><IsFactorBased></td><td>Boolean</td><td>True selected via groups are snapped by a maximum area equal to the tolerance (i.e., <Snaptol> and <AlignTol>) times the surface area of the via group.
False selected via groups are snapped by a maximum distance of the tolerance.</td></tr>
<tr><td><SnapTol></td><td>Integer</td><td>Vias at or within this tolerance are snapped.</td></tr>
<tr><td><AlignTol></td><td>Integer</td><td>Vias at or within this tolerance are aligned.</td></tr>
<tr><td><RemoveDanglingVias></td><td>Boolean</td><td>True removes vias from selection that have no connectivity or have connectivity only to a single layer.
False deactivates this check.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SnapVias([<NAME:vias>, <Objects>], [<NAME:Setting Contents>, <IsFactorBased>, <SnapTol>, <AlignTol>, <RemoveDanglingVias>])

Python Example	`oEditor.SnapVias(`
	`[`
	`  "NAME:vias",`
	`  "via_0",`
	`  "via_1",`
	`  "via_2"`
	`],`
	`[`
	`  "NAME:Setting Contents",`
	`  "IsFactorBased:=", True,`
	`  "snapTol:=", "3",`
	`  "AlignTol:=", "300um",`
	`  "removeDanglingVias:=", True`
	`])`

SnapViasOnLayer (IC Editor)

Forces the edges of selected vias to snap to alignment on the target layer.

UI Access	From the **Layers** window, right-click the target layer and select **Snap Vias**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Layer_Name></td><td>String</td><td>Name of the target layer.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Setting Contents>
<IsFactorBased>
<SnapTol>
<AlignTol>
<RemoveDanglingVias></td></tr>
<tr><td>NAME:Setting Contents</td><td>Array</td><td>Required first parameter which must have the specified value (i.e., "Setting Contents").</td></tr>
<tr><td>IsFactorBased</td><td>Array</td><td>True selected via groups are snapped by a maximum area equal to the tolerance (i.e., <Snaptol> and <AlignTol>) times the surface area of the via group.
False selected via groups are snapped by a maximum distance of the tolerance.</td></tr>
<tr><td>SnapTol</td><td>Array</td><td>Vias at or within this tolerance are snapped.</td></tr>
<tr><td>AlignTol</td><td>Array</td><td>Vias at or within this tolerance are aligned.</td></tr>
<tr><td>RemoveDanglingVias</td><td>Array</td><td>True removes vias from selection that have no connectivity or have connectivity only to a single layer.
False deactivates this check.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	SnapVias(<Layer_Name>, [<NAME:Setting Contents>, <IsFactorBased>, <SnapTol>, <AlignTol>, <RemoveDanglingVias>])
Python Example	`oEditor.SnapViasOnLayer(` `["Trace",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "NAME:Setting Contents",
  "IsFactorBased:=", False,
  "snapTol:=", "5mm",
  "AlignTol:=", "5mm",
  "removeDanglingVias:=", True
])
```

UnitePrimitivesOnLayer (IC Editor)

Unites primitives on the target layer.

UI Access	From the **Layers** window, right-click the target layer and select **Unite Primitives**.
Parameters	Name | Type | Description <*NAME*> | String | Name of the target layer.
Return Value	None.

Python Syntax	UnitePrimitivesOnLayer(<*NAME*>)
Python Example	`oEditor.UnitePrimitivesOnLayer("Trace")`

WrapGeometries (IC Editor)

Wraps the selected polygons.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	After choosing the appropriate geometries, select **Wrap Geometry** from the **Layout** ribbon.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:Contents> <Elements> <Objects> </td></tr><tr><td><*NAME:Contents*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Contents").</td></tr><tr><td><*Elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Elements").</td></tr><tr><td><*Objects*></td><td>Array</td><td>A structured array containing a comma-separated list of integers, where each integer represents a selected polygon(i.e., "rect_2577", "rect_2578" , "ect_2579", etcetera)</td></tr></table>
Return Value	None.

Python Syntax	WrapGeometries ([<NAME:Contents>, Elements, Objects])

Python Example	`oEditor.WrapGeometries(`
	`[`
	`  "NAME:Contents",`
	`  "elements:=", ["rect_2577", "rect_2578", "rect_2579"]`
	`])`

WrapGeometry (IC Editor)

Sets a tolerance and wraps polygons beginning with the selected polygon.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	From the **Layout** ribbon, select **Wrap Geometry**. Right-click the selected geometry within the design and Adjust the tolerance slider as necessary. Then click **Wrap**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SEED_OBJECT_NAME*></td><td>String</td><td>Name of the selected polygon.</td></tr><tr><td><*Snap_Dis_Tolerance*></td><td>Integer</td><td>Polygons at or within this range are wrapped.</td></tr></table>
Return Value	None.

Python Syntax	WrapGeometry (<*SEED_OBJECT_NAME*>, <*Snap_Dis_Tolerance*>)
Python Example	`oEditor.WrapGeometry("rect_1855", "5um")`

Layers Methods

The following methods are available to manipulate layers:

AddLayer (Layout Editor)

AddObject (IC Editor and Layout Editor)

AddStackupLayer (Layout Editor)

ChangeLayer (Layout Editor)

ChangeLayers (Layout Editor)

ClearLayerMappings (Layout Editor)

CreateBoardBend (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DuplicateAcrossLyrs (Layout Editor)

ExportStackupXML (Layout Editor)

GetAllLayerNames (Layout Editor)

GetLayerInfo (Layout Editor)

GetStackupLayerNames (Layout Editor)

ImportStackupXML (Layout Editor)

ProcessBentModelCmd (Layout Editor)

RemoveLayer (Layout Editor)

SetLayerMapping (Layout Editor)

SyncRigidFlexSandbox (Layout Editor)

AddObject (Layout Editor and IC Editor)

Adds a user-selected object to the active design.

<table>
<tr><td>UI Access</td><td>Not strictly applicable, although objects can be drawn by making the appropriate selection from the Layout ribbon (i.e., Draw polygon, Draw line, Draw rectangle, Draw arc, Draw circle.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ObjectType></td><td>String</td><td>Type of object to add (i.e., rect (i.e., rectangle), circle, poly (i.e., polygon), line, or 3D line).</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing the parameters required for the type of object selected. The parameters differ for each object. Refer to the following pages:
CreateRectangle (Layout Editor)
CreateCircle (Layout Editor)</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• CreatePolygon (Layout Editor) • CreateLine (Layout Editor) • Create3DLine (Layout Editor) • CreatePartitionLine (IC Editor)
Return Value	Returns the name of the newly created object (**IC Mode** only).		

Python Syntax	AddObject(<ObjectType>, [<Parameters>])

Python Example	`oEditor.AddObject("rect",`
	`  [`
	`  "NAME:Contents",`
	`  "rectGeometry:=", [`
	`  "Name:=", "rect_5",`
	`  "LayerName:=", "Trace",`
	`  "lw:=", "0",`
	`  "Ax:=", "-84mm",`
	`  "Ay:=", "16mm",`
	`  "Bx:=", "-77mm",`
	`  "By:=", "12mm"]`
	`  ])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddLayer (Layout Editor)

Adds a layer in a layout definition. This command can take all parameters available in ChangeLayers (Layout Editor) but cannot be recorded.

Note: As with other Layout scripting interface commands that modify the layout, this command is not intended for use within scripts that define footprints. The command behavior from within such a script is undefined and may be unexpected. Use the LayoutHost scripting interface commands within scripts that define footprints.

<table>
<tr><td>UI Access</td><td colspan="3">None</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><LayerArray></td><td>Array</td><td>An array that contains these parameters in order:
<Layername>
<Type>
<TopBottom>
<Color>
<Transparency>
<Pattern>
<VisFlag>
<Locked>
<Draw></td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of layer.</td></tr>
<tr><td><Type></td><td>String</td><td>Type of layer, such as "user", "soldermask", "solderpaste", "silkscreen", "assembly", "wirebond", "glue", "user", "Postprocessing", or "outline".</td></tr>
<tr><td><TopBottom></td><td>String</td><td>Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".</td></tr>
<tr><td><Color></td><td>Integer</td><td>Optional. A code to indicate the color of the layer. It is an RGB code in the hex format bbggrr. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.</td></tr>
<tr><td><Transparency></td><td>Integer</td><td>Optional. A percentage that indicates the transparency of the layer between 0</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			(opaque) and 100 (completely transparent and not visible).
	<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
	<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
	<Locked>	Boolean	Optional. Whether the layer is locked.
	<Draw>	Integer	Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
Return Value	None		

Python Syntax	oEditor.AddLayer(["NAME:layers", "Name:=", "<LayerName>", "Type:=", "<Type>",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Top Bottom:=", "<TopBottom>", "Color:=", <Color>, "Transparency:=", <Transparency>, "Pattern:=", <Pattern>, " VisFlag:=" , <VisFlag>, "Locked:=", <Locked>, "DrawOverride:=", <Draw>,])
Python Example	oEditor = oDesign.SetActiveEditor("Layout") oEditor.AddLayer(["NAME:layers", "Name:=","signal", "Type:=", "assembly", "Top Bottom:=", "neither", "Color:=", 16711680, "Transparency:=", 0, "Pattern:=", 1, "VisFlag:=", 127, "Locked:=", False, "DrawOverride:=", 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

AddStackupLayer (Layout Editor)

Adds a stackup layer in the Layout editor. This command can take all parameters available in ChangeLayers (Layout Editor) but cannot be recorded.

UI Access	None		
Parameters	Name	Type	Description

	<LayerArray>	Array	An array that contains these parameters in order:
			• <Layername>
			• <Type>
			• <TopBottom>
			• <Color>
			• <Transparency>
			• <Pattern>
			• <VisFlag>
			• <Locked>
			• <Draw>
			• and an array with the parameters:
			  • <Thickness>
			  • <LowerElevation>
			  • <Material>

	<LayerName>	String	Name of layer.
	<Type>	String	Type of layer, either "signal" or "dielectric".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<TopBottom>	String	Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
	<Color>	Integer	Optional. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
	<Transparency>	Integer	Optional. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
	<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
	<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
	<Locked>	Boolean	Optional. Whether the layer is locked.
	<Draw>	Integer	Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
	<Thickness>	String	Thickness of the layer, contains the measurement and units.
	<LowerElevation>	String	The elevation of the lower edge of the layer, contains the measurement and units.
	<Material>	String	The material the layer is made of.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	oEditor.AddStackupLayers(["NAME:stackup layer", "Name:=", "<LayerName>", "Type:=", "<Type>", "Top Bottom:=", "<TopBottom>", "Color:=", <Color>, "Transparency:=", <Transparency>, "Pattern:=", <Pattern>, "VisFlag:=" , <VisFlag>, "Locked:=", <Locked>, "DrawOverride:=", <Draw>, ["NAME:sublayer", "Thickness:=", "<Thickness>", "LowerElevation:=", "<LowerElevation>", "Material:=", "<Material>",]])
Python Example	`oEditor = oDesign.SetActiveEditor("Layout")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oEditor.AddStackupLayer(
  [
    "NAME:stackup layer",
    "Name:=", "Cover",
    "Type:=", "signal",
    "Top Bottom:=", "neither",
    "Color:=", 16711680,
    "Transparency:=", 0,
    "Pattern:=", 1,
    "VisFlag:=", 127,
    "Locked:=", False,
    "DrawOverride:=", 0,
    [
      "NAME:Sublayer",
      "Thickness:=", "0mil",
      "LowerElevation:=", "20mil",
      "Material:=", "gold",
    ]
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeLayer (Layout Editor)

Changes one or more attributes or parameters of a single layer.

> **Note:** To execute the command, **all** of the following parameters must be entered in the command line even if the user only intends to change one attribute.

<table>
<tr><td>UI Access</td><td>Limited to showing/hiding element and object visibility in the Layers window.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><LayerArray></td><td>Array</td><td>An array that contains these parameters in order:
<mode>
<Layername>
<ID>
<Type>
<TopBottom>
<Color>
<Transparency>
<Pattern>
<VisFlag>
<Locked>
<Draw>
<Subarray></td></tr>
<tr><td><mode></td><td>String</td><td>Type of stackup. Can be "Laminate","Multizone", or "Overlap" .</td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of layer.</td></tr>
<tr><td><ID></td><td>Integer</td><td>A numerical identification for the layer.</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<Type>	String	Type of layer, such as "user", "signal", "dielectric", "soldermask", "solderpaste", "silkscreen", "assembly", "wirebond", "dielectric", "glue", "user", "Post-processing", or "outline".
<TopBottom>	String	*Optional*. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
<Color>	Integer	*Optional*. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
<Transparency>	Integer	*Optional*. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
<Pattern>	Integer	*Optional*. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
<Locked>	Boolean	*Optional*. Whether the layer is locked.
<DrawOverride>	Integer	*Optional*. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
<Zones>		
<Subarray>	Array	*Optional*. An array that contains these parameters in order: • <Thickness> • <LowerElevation>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Roughness> • <BotRoughness> • <SideRoughness> • <Material> • <FillMaterial> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<Thickness>	String	Thickness of the layer, contains the measurement and units.
	<LowerElevation>	String	The elevation of the lower edge of the layer, contains the measurement and units.
	<Roughness>	String	The roughness of the top of the layer, contains the measurement and units.
	<BotRoughness>	String	The roughness of the bottom of the layer, contains the measurement and units.
	<SideRoughness>	String	The roughness of the side of the layer, contains the measurement and units.
	<Material>	String	The material the layer is made of.
	<FillMaterial>	String	The material to use to fill in around geometry on the layer.
Return Value	None		

Python Syntax	oEditor.ChangeLayer(<LayerArray>)

Python Example	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.ChangeLayer(`
	`  ["NAME:stackup layer",`
	`  "Name:=" , "MQ",`
	`  "ID:=" , 20,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Type:="        , "signal",
    "Top Bottom:="  , "neither",
    "Color:="       , 6136303,
    "Transparency:="    , 60,
    "Pattern:="         , 1,
    "VisFlag:="         , 96,
    "Locked:="          , False,
    "DrawOverride:="    , 0,
    "Zones:="           , [],
      ["NAME:Sublayer",
        "Thickness:="       , "0.745um",
        "LowerElevation:="  , "737.155um",
        "Roughness:="       , "0um",
        "BotRoughness:="    , "0um",
        "SideRoughness:="   , "0um",
        "Material:="        , "MQ",
        "FillMaterial:="    , "FR4_epoxy"
      ]
    ]
  )
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClearLayerMappings (Layout Editor)

Clears all layer mappings associated with the specified component.

UI Access	None
Parameters	Name | Type | Description <ComponentName> | String | Name of the component whose layer mappings will be cleared.
Return Value	None

Python Syntax	ClearLayerMappings(<componentName>)

Python Example	`ClearLayerMappings(`
	`    "Component_A"`
	`)`

DuplicateAcrossLyrs (Layout Editor)

Use: Duplicate selected objects (layout and footprint) to other layers.

Command: Draw > Duplicate > Across Layers

Syntax: DuplicateAcrossLyrs Array("NAME:elements", "element-name",…), Array("NAME:layers", "layer-name",…)

Return Value: None

Parameters: `<element-name> // The name of the element to be duplicated.`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<*layer-name*> // The name of the layer to duplicate elements to.

ExportGDSII (Layout Editor)

Exports results to GDSII.

<table>
<tr><td>UI Access</td><td>File > Export > GDSII.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><*OptionParameters*></td><td>Array</td><td>Structured array.

["NAME:options",

"FileName:=", <string, GDSII export file name.>,

"NumVertices:=", <integer, maximum number of vertices allowed in a polygon (specify zero if unlimited).>,

"ArcTol:=", <double, arc tolerance (in meters); the value used to discretize arcs. Smaller the value the greater the number of edges exported. For 0.1mm use 0.0001.>,

"LayerMap:=", [

"entry:=", [

"layer:=", <string, name of a layer to export.>,

"id:=", <integer, GDSII layer ID to assign the exported layer.>,</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"include:=", <boolean, if true, the layer is exported.>]]]

Return Value	None.

Python Syntax	ExportGDSII(<*OptionParameters*>)
Python Example	oEditor.ExportGDSII(["NAME:options", "FileName:=", "C:/Projects/ design.gds", "NumVertices:=", 0, "ArcTol:=", 2E-006, "LayerMap:=", ["entry:=",["layer:=", "Bottom", "id:=", 2, "include:=", true], "entry:=", ["layer:=", "Top", "id:=", 1,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"include:=", true]]]

ExportStackupXML (Layout Editor)

Exports the stackup in XML format.

UI Access	From the **Edit Layers** window, select **Stackup** > **Export XML** to open an explorer window. Navigate to an appropriate directory, choose a **File name**, and click **Save**.
Parameters	Name | Type | Description <*SaveAs*> | String | User-specified directory in which to save the new stackup XML file.
Return Value	None.

Python Syntax	ExportStackupXML(<*SaveAs*>)
Python Example	`oEditor.ExportStackupXML("C:\\Users\\USERNAME\\Downloads\\test.xml")`

GetAllLayerNames (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetAllLayerNames

Return Value: Array of strings which are the names of all layers in the layout, blackbox, or footprint.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetLayerInfo (Layout Editor)

Retrieves information on a specified layer in the Layout editor.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of the layer</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>
An array of strings, as follows:

Type: type of name
TopBottomAssociation: "Top"|"Neither"|"Bottom"|"Template"|"Invalid"
Color: integer [representing rgb in hex]
IsVisible: "true"|"false". This is true if any these types of objects are visible:

IsVisibleShape: "true"|"false"[for stackup layer only]
IsVisiblePath: "true"|"false"[for stackup layer only]
IsVisiblePad: "true"|"false"[for stackup layer only]
IsVisibleHole: "true"|"false"[for stackup layer only]
IsVisibleComponent: "true"|"false"[for stackup layer only]

IsLocked: "true"|"false"
LayerId: integer [the ID for the layer]

The following are also in the array if the layer is a stackup layer:

Index: integer [the stackup order index]
LayerThickness: double [total layer thickness, in meters]
EtchFactor: double [This will not appear if the layer has no etch factor defined.]
IsIgnored: "true"|"false"

</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• NumberOfSublayers: 1 [This is always 1.]
	• Material0: material name
	• FillMaterial0: material name
	• Thickness0: expression with units
	• LowerElevation0: expression with units
	• If roughness is defined, some or all of the following appear:
	  • Roughness0Type: Groiss | Huray
	  • Roughness0: value and units or surface ratio value for Huray
	  • BottomRoughness0 Type: Groiss | Huray
	  • BottomRoughness0: value and units or surface ratio value for Huray
	  • SideRoughness0 Type: Groiss | Huray
	  • SideRoughness0: value and units or surface ratio value for Huray
	For example:
	`['Type: signal', 'TopBottomAssociation: Neither', 'Color: 32512d', 'IsVisible: true', ' IsVisibleShape: true', ' IsVisiblePath: true', ' IsVisiblePad: true', ' IsVisibleHole: true', ' IsVisibleComponent: true', 'IsLocked: false', 'LayerId: 7', 'Index: 0', 'LayerThickness: 0', 'IsIgnored: false', 'NumberOfSublayers: 1', 'Material0: copper', 'FillMaterial0: FR4_epoxy', 'Thickness0: 0mil', 'Lower-Elevation0: 62mil', 'Roughness0 Type: Huray', 'Roughness0: 0mil, 2.9', 'Bot-tomRoughness0 Type: Huray', 'BottomRoughness0: 0mil, 2.9', 'SideRoughness0 Type: Huray', 'SideRoughness0: 0mil, 2.9']`

Python Syntax	GetLayerInfo(<layer_name>)
Python Example	`oEditor = oDesign.SetActiveEditor("Layout")` `oEditor.GetLayerInfo("TopLayer")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetStackupLayerNames (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetStackupLayerNames

Return Value: array of strings which are the names of all layers in the layout, blackbox, or footprint.

Parameters: None

ImportStackupXML (Layout Editor)

Imports an XML file into an active project.

UI Access	From the **Edit Layers** window, select **Stackup** > **Import XML**.
Parameters	Name | Type | Description <*XMLFileName*> | String | Full path to XML file.
Return Value	None.

Python Syntax	ImportStackupXML(<*XMLFileName*>)

Python Example	`oProject = oDesktop.SetActiveProject("Project_Name")`
	`oDesign = oProject.SetActiveDesign("DesignName")`
	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.ImportStackupXML("C:/Files/example.xml")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveLayer (Layout Editor)

UI Access	From the **Layers** window, right-click the target layer and select **Remove**.

General Mode

Removes the target layer or target stackup layer.

Parameters	Name	Type	Description
	<LayerName>	String	Name of the target layer.
Return Value	None.		

Python Syntax	RemoveLayer(<LayerName>)
Python Example	`oEditor.RemoveLayer("Ground")`

IC Mode (IC Editor)

Removes all objects from the target layer or remove all objects *and* the target layer.

Parameters	Name	Type	Description
	<Parameters>	Array	Structured array containing these parameters in order: • <NAME:Contents>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <NAME> • <RemoveLayer>
	<NAME:Contents>	String	Required first parameter which must have the specified value (i.e., "Contents").
	<LayerName>	String	Name of the target layer.
	<RemoveLayer>	Boolean	**True** removes the layer and all objects on the target layer. **False** removes only all objects.
Return Value	None.		

Python Syntax	RemoveLayer([*<NAME:Contents>*, <LayerName>], <RemoveLayer>)

Python Example	`oEditor.RemoveLayer(`
	`  [`
	`  "NAME:Contents",`
	`  "Ground"`
	`  ], False)`

SetLayerMapping (Layout Editor)

Use: Set layer mapping.

Command: None.

Syntax: SetLayerMapping <component name>, <design layer>, <footprint layer>

Return Value: None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <component name> is a string that specifies the name of the component

<design layer> is a string that specifies the name of the design layer

<footprint layer> is a string that specifies the name of the footprint layer

Example: oEditor.SetLayerMapping "2", "Bottom Signal", "Top"

SyncRigidFlexSandbox (Layout Editor)

Create or modify geometry and define zones and rigid-flex bends in a design. Existing elements not included in the argument are deleted.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><*Contents*></td><td>Array</td><td>Structured Array containing:
["Name:contents",
<*Geometry1Array*>,<*Geometry2Array*>,<*Geometry3Array*>...]
Note: The number and order of geometries and bends will differ for each design and at the user's discretion. The order of zones in the array implies the Zone Order parameter in Electronics Desktop.</td></tr>
<tr><td><*Geometry*></td><td>Array</td><td>Structured Array containing:
["Name:=", <string zone# name>,
"LayerNameInfo:=", #list of strings defining layer</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			properties] **Note:** Depending on the type of geometry ((i.e., polygon, rectangle, circle, line, or board bend), additional parameters may be applicable. For example, adding "`Is Fixed:=true` within a <*Geometry*> array will assign the **Fixed** status to the primitive. The primitive will then become a fixed reference position during the bending process. Fixing more than one primitive in a design is outside the scope of this function and results are therefore unpredictable.
Return Value	None.		
Python Syntax	SyncRigidFlexSandbox (<Contents>, <Geometry>...)		
Python Example			

Net and NetClass Methods

The following net and net class primitives are available:

CreateDifferentialPair

CreateNetClass

DeleteNets

DelNetClass

GetNetClasses

GetNetClassNets

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNetConnections

GetNets

HighlightNet

ModifyDifferentialPair

ModifyNetClass

RenameNet

ResetDifferentialPairs

SelectNetConnected

SelectNetConnected

SetNetColor

SetNetVisible

ToggleNetHighlight

CreateDifferentialPair (Layout Editor)

Creates a differential pair from two target nets.

UI Access	From the **Nets** window, **Ctrl+click** two nets from the list. Then right-click within the **Nets** window and select **Create Differential Pair**.
Parameters	

Name	Type	Description
<*DiffPairName*>	String	Name of the new differential pair being created (e.g., DiffPair1)
<*Description*>	String	Redundant parameter, intentionally left blank.
<*Parameters*>	Array	Array containing two target net names:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Net1> • <Net2>
	<Net1>	String	One of two target nets used to create the new differential pair (e.g., GND).
	<Net2>	String	One of two target nets used to create the new differential pair (e.g., Port1).
Return Value	None		

Python Syntax	CreateDifferentialPair(<DiffPairName>, <Description>, [<Net1>, <Net2>])
Python Example	`oEditor.CreateDifferentialPair("DiffPair1", "", ["GND", "Port1"])`

CreateNetClass

*Use:*Create a new net class into a layout.

*Command:*CreateNetClass.

*Syntax:*CreateNetClass (<name> <description> <nets name list>)

*Return Value:*None

DeleteNets (Layout Editor)

Removes specified nets from the project.

UI Access	None.		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<netName>	String	Name of unwanted net.
Return Value	None.		

Python Syntax	DeleteNets ([<*netName*>, ...])

Python Example	`oEditor.DeleteNets(`
	`  [`
	`  "NET_1",`
	`  "NET_2",`
	`  "NET_3"`
	`  ])`

DelNetClass

*Use:*Delete a net class in a layout.

*Command:*DelNetClass

*Syntax:*DelNetClass (<names>)

*Return Value:*None

GetNetClasses

*Use:*Gets all the net classes in a layout.

*Command:*GetNetClasses

*Syntax:*GetNetClasses ()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

*Return Value:*None

GetNetClassNets

*Use:*Gets the list of nets in a net class.

*Command:*GetNetClassNets

*Syntax:*GetNetClassNets (<name>)

*Return Value:*None

GetNetConnections (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetNetConnections(<netName>)

Return Value: Array of strings containing the connections for the net identified by the netName argument. The strings in the array are in one of four formats:

"ComponentPin compID pinname x y EdgePort"

"ComponentPin compID pinname x y PinPadstack: padstackName"

"InterfacePort portname portID x y Padstack: padstackName"

"EdgePort portname portID EdgeInfo: Primitive id, edge index[Primitive id, edge index]"

where *compID* is the component instance identifier, *pinname* is the name of the connected pin, *x* and *y* are the connection point, *padstackName* is the name of the padstack, *portname* is the name of the port, *portID* is the identifier for the interface

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

port, and *id* and *index* identify edges involved in an edgeport.

Parameters: <netName>

Type: String

The name of the net.

GetNets (Layout Editor)

Returns the names of all nets in the active design.

UI Access	N/A.
Parameters	None.
Return Value	Array of the names of the nets in the active design, separated by commas, or None if no nets exist.

Python Syntax	GetNets()
Python Example	`oEditor.GetNets()` `['gnd', 'net1', 'net2']`

HighlightNet (Layout Editor)

Highlights or removes the highlight from all elements in a net.

UI Access	None.
Parameters	Name | Type | Description <*requiredPara-* | String | Required first parameter which must have the value

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	meterKeyword>		"Name:Args".
	<*requiredNetKeyword*>	String	Required second parameter which must have the value "Name:=". Must be the first parameter for each net.
	<*netName*>	String	Name of the net.
	<*requiredFlagKeyword*>	String	Required fourth parameter which must have the value "Hi:=". Must be the third parameter for each net.
	<*highlightFlag*>	Boolean	True if the net should be highlighted. False if it should not.
Return Value	None.		

Python Syntax	HighlightNet (["Name:Args", "Name:=", " <*netName*>", "Hi:=", <*highlightFlag*>, ...])

Python Example	`oEditor.HighlightNet(`
	`  [`
	`  "NAME:Args",`
	`  "Name:=", "net_0",`
	`  "Hi:=", True`
	`  ])`

ModifyDifferentialPair (Layout Editor)

Modifies a differential pair from two target nets.

UI Access	From the **Nets** window, select the **Differential** tab. Then right-click a differential pair and select **Swap**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*DiffPairName*></td><td>String</td><td>Name of the target differential pair (i.e., Sig_0).</td></tr><tr><td><*NewName*></td><td>String</td><td>New name of the target differential pair (i.e., since it will be unchanged, Sig_0).</td></tr><tr><td><*Description*></td><td>String</td><td>Redundant parameter, intentionally left blank.</td></tr><tr><td><*Parameters*></td><td>Array</td><td>Array containing the nets in the differential pair in their new swapped order: <Net1> <Net2> </td></tr><tr><td><*Net1*></td><td>String</td><td>A net in the differential pair, presumably now in the swapped position (e.g., Sig_P).</td></tr><tr><td><*Net2*></td><td>String</td><td>A net in the differential pair, presumably now in the swapped position (e.g., Sig_N).</td></tr></table>
Return Value	None

Python Syntax	ModifyDifferentialPair(<DiffPairName>, <NewName>, <Description>, [<Net1>, <Net2>])
Python Example	`oEditor.ModifyDifferentialPair("Sig_0", "Sig_0", "", ["Sig_P", "Sig_N"])`

ModifyNetClass

*Use:*Modify an existing net class in a layout.

*Command:*ModifyNetClass

*Syntax:*ModifyNetClass (<name> <new name> <new description> <new nets name list>)

*Return Value:*None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ResetDifferentialpairs (Layout Editor)

Resets a differential pair to its original state.

<table>
<tr><td>UI Access</td><td colspan="3">From the Nets window, select the Differential tab. Then right-click a differential pair and select Edit.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters1></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:New Differential Pairs>
<DiffPair1>
<DiffPair2></td></tr>
<tr><td><NAME:New Differential Pairs></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "New Differential Pairs").</td></tr>
<tr><td><DiffPair1></td><td>Array</td><td>Array containing the following parameters for the target differential pair in order:
<DiffPairName>
<Description1>
<DiffPair1Nets></td></tr>
<tr><td><DiffPair1Name></td><td>String</td><td>Name of the target differential pair (i.e., 1).</td></tr>
<tr><td><Description1></td><td>String</td><td>Redundant parameter, intentionally left blank.</td></tr>
<tr><td><DiffPair1Nets></td><td>Array</td><td>Array containing the nets in the target differential pair:
<Net1>
<Net2></td></tr>
<tr><td><Net1></td><td>String</td><td>The first net in the target differential pair (e.g., Port1).</td></tr>
<tr><td><Net2></td><td>String</td><td>The second net in the target differential pair (e.g., Sig_P).</td></tr>
<tr><td><DiffPair2></td><td>Array</td><td>Array containing the following parameters for the target differential pair in order:
<DiffPairName></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Description2> • <DiffPair2Nets>
	<DiffPair2Name>	String	Name of the target differential pair (i.e., 2).
	<Description2>	String	Redundant parameter, intentionally left blank.
	<DiffPair2Nets>	Array	Array containing the nets in the target differential pair: • <Net3> • <Net4>
	<Net3>	String	The first net in the target differential pair (e.g., Sig_N).
	<Net4>	String	The second net in the target differential pair (e.g., Port2).
Return Value	None		

Python Syntax	ResetDifferentialPairs([<NAME:New Differential Pairs>, [<DiffPair1>, <Description1>, [<Net1>, <Net2>]], [<DiffPair2>, <Description2>, [<Net3>, <Net4>]]])

Python Example	`oEditor.ResetDifferentialPairs(`
	`  [`
	`  "NAME:New Differential Pairs",`
	`  [`
	`  "1",`
	`  "",`
	`  ["Port1", "Sig_P"]`
	`  ],`
	`  [`
	`  "2",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	```"", ["Sig_N", "Port2"]]])```

SelectNetConnected (Layout Editor)

Select all elements in the active design connected to the target net.

UI Access	From the **Layout** tab, right-click an object in the active design and select **Nets** > **Select Net Connected**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:elements> <TargetNet> </td></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "elements").</td></tr><tr><td><*TargetNet*></td><td>String</td><td>Name of the target net.</td></tr></table>
Return Value	None.

Python Syntax	SelectNetConnected([<NAME:elements>, <TargetNet>])
Python Example	`oEditor.SelectNetConnected(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:elements", "circle_1288",])

SelectPhysicallyConnected (Layout Editor)

Selects all elements physically connected to the target object.

UI Access	From the **Layout** tab, right-click an object in the active design and select **Nets** > **Select Physically Connected**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: • <NAME:elements> • <TargetObject></td></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "elements").</td></tr><tr><td><*TargetObject*></td><td>String</td><td>Name of the target object.</td></tr></table>
Return Value	None.

Python Syntax	SelectPhysicallyConnected([<NAME:elements>, <TargetObject>])
Python Example	oEditor.SelectPhysicallyConnected(["NAME:elements",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"circle_1288",])

SetNetColor (Layout Editor)

Sets the color of a target net.

UI Access	From the **Nets** window, right-click a net and select **List** to open the **Design List** window. From any row, double-click the bar in the **Color** column to open the **Color** window. Select a new color, then click **OK** to change the net's color.
Parameters	

Name	Type	Description
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:Args> • <Name> • <Color>
<*NAME:Args*>	String	Required first parameter which must have the specified value (i.e., "Args").
<*Name*>	String	Name of the target net.
<*Color*>	Array	Structured array containing the RGB (i.e., Red, Green, Blue) color code, which the target net will be set to: • <R> • <G> • <B> • <Check_Containment> • <Via_Persistence>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<R>	Integer	Value of red color.
	<G>	Integer	Value of green color.
	<B>	Integer	Value of blue color.
Return Value	None.		

Python Syntax	SetNetColor([<NAME:Args>, <Name>, <Color>, [<R>, <G>, <B>]])

Python Example	`oEditor.SetNetColor(`
	`  [`
	`  "NAME:Args",`
	`  "Name:=", "Net_1",`
	`  "Color:=", ["R:=", 0, "G:=", 0, "B:=", 160]`
	`  ])`

SetNetVisible (Layout Editor)

Shows and highlights or hides specified nets in the Layout Editor. Multiple nets can be highlighted or hidden at once.

UI Access	On the **Nets** tab of the **Nets** window, highlight and right click on one or more nets. Click **Hide** or any of the **Show** options.		
Parameters	Name	Type	Description
	<requiredPara-meterKeyword>	String	Required first parameter which must have the value "Name:Args".
	<requiredNameKeyword>	String	Required second parameter which must have the value

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Name:=". Must be the first parameter for each net.
	<netName>	String	Name of the net
	<requiredFlagKeyword>	String	Required fourth parameter which must have the value "Vis:=". Must be the third parameter for each net.
	<visibilityFlag>	Boolean	True if the net should be visible. False if it should be hidden.
Return Value	None.		

Python Syntax	SetNetVisible (["Name:Args", "Name:=", "<*netName*>", "Vis:=", <*highlightFlag*>, ...])

Python Example	`oEditor.SetNetVisible(`
	`  [`
	`  "NAME:Args",`
	`  "Name:=", "Net_1",`
	`  "Vis:=", True`
	`  ])`
	`oEditor.SetNetVisible(`
	`  [`
	`  "NAME:Args",`
	`  "Name:=", "<NO-NET>",`
	`  "Vis:=", False,`
	`  "Name:=", "Ground",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Vis:=", False,
    "Name:=", "Chassis",
    "Vis:=", True
  ])
```

ToggleNetHighlight (Layout Editor)

Highlights or removes the highlight from all objects in the nets of the target elements.

UI Access	After selecting one or more elements within the **Layout Editor**, navigate to **Layout** > **Nets** > **Toggle Net Highlight**.
Parameters	See table below
Return Value	None.

Name	Type	Description
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:elements> • <Names>
<*NAME:elements*>	String	Required first parameter which must have the specified value (i.e., "elements").
<*Names*>	String	Names of the target elements to highlight, separated by commas.

Python Syntax	ToggleNetHighlight([<NAME:elements>, <Names>])
Python Example	See code below

```
oEditor.ToggleNetHighlight(
  [
    "NAME:elements",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"rect_0", "rect_2", "circle_1" "R96-2"])

Port Methods

The following port methods are available:

AddCircuitRefPort

AddPortsToAllNets

AddPortsToNets

AddRefPort

AlignPorts

ClearRefPort

CreateCircuitPort

CreateEdgePort

CreateNPort

CreatePortInstancePorts

CreatePortsOnComponents

CreatePortsOnComponentsByNet

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPortInfo

RemovePort

RemovePortsFromAllNets

RemovePortsFromNet

RemovePortsOnComponents

SetHfssExtentsVisible

SetHfssPortsVisible

AddCircuitRefPort (Layout Editor)

Assigns an edge as a reference to an existing port, modeled as a circuit port.

UI Access	None

Parameters

Name	Type	Description
<NAME:Edge>	String	Object descriptor header.
<Et>	String	Edge type identifier ("pe" for primitive edge, "pse" for via edge).
<Prim>	String	Name of the primitive (used with "pe").
<Edge>	Integer	Edge number on the primitive.
<Sel>	String	Name of the via to use (used with "pse").
<Layer>	Integer	ID of the layer containing the via pad.
<Sx>	Integer	Start X coordinate of the edge arc.
<Sy>	Integer	Start Y coordinate of the edge arc.
<Ex>	Integer	End X coordinate of the edge arc.
<Ey>	Integer	End Y coordinate of the edge arc.
<H>	Integer	Height of the arc (0 for straight edge).
<Rad>	Integer	Arc size in radians (0 for straight edge).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	AddCircuitRefPort([<*NAME:Edge*>, [<*Et*>, <*Sel*>, <*Layer*>, <*Sx*>, <*Sy*>, <*Ex*>, <*Ey*>, <*H*>, <*Rad*>]])

Python Example	`oEditor.AddCircuitRefPort(`
	`  [`
	`  "NAME:Edge",`
	`  [`
	`  "Et:=", "pse",`
	`  "Sel:=", "Via1",`
	`  "Layer:=", 3,`
	`  "Sx:=", "1.0mm",`
	`  "Sy:=", "2.0mm",`
	`  "Ex:=", "3.0mm",`
	`  "Ey:=", "4.0mm",`
	`  "H:=", "0mm",`
	`  "Rad:=", "0"`
	`  ]`
	`  ]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddPortsToAllNets (Layout Editor)

Adds ports to all the pins on all nets.

UI Access	From the **Layout** tab > **Nets**, right-click and click **Create Ports**.
Parameters	None
Return Value	None

Python Syntax	AddPortsToAllNets()
Python Example	`oEditor.AddPortsToAllNets()`

AddPortsToNet (Layout Editor)

Adds the selected ports to the current selected net.

UI Access	Right-click within the **Layout Editor** and select **Port** > **Create Ports on Net**.
Parameters	Name / Type / Description *<Parameters>* / Array / Structured array containing these parameters in order: • <NAME:Nets> • <net_name> *<NAME:Nets>* / String / Required first parameter which must have the specified value (i.e., "Nets"). *<port_name>* / String / A comma-separated list of port names (e.g., "port1", "port2", "port3", et cetera).
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddPortsToNet([<*NAME:Nets*>, <*net_name*>])

Python Example	`oEditor.AddPortsToNet(`
	`  [`
	`  "NAME:Nets",`
	`  "port1",`
	`  "port2"`
	`  ])`

AddRefPort (Layout Editor)

Creates a reference port from edges and associates the reference port with each of the named ports.

UI Access	Right-click within the **Layout Editor** and select **Port** > **Create**.		
Parameters	Name	Type	Description
	<*Port_List*>	Array	Structured array containing one or more *Port_Name* strings. Each *Port_Name* represents a named port (e.g., "port1", "port2", "port3").
	<*Port_Name*>	String	Individual port name string used within the *Port_List* array.
	<*Edge_Array*>	Array	Structured array containing these parameters in order: • <NAME:Contents> — Required string with value "Contents" • <Edge> — One entry per *Port_Name* in *Port_List* Each *Edge* entry may reference a predefined edge object

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		(e.g., "edge_description") or contain a full *Edge_Description* array inline.
<*NAME:Contents*>	String	Required first parameter which must have the specified value (i.e., "Contents").
<*Edge*>	Array	Structured array containing an <Edge_description> parameter for each <Port_Name> parameter provided in the <Port List> array.
<*Edge_ Description*>	Array	Name of the array containing the geometry of the selected edge, containing these parameters in order: • <Layer> • <Port> • <Et> • <Pe> • <Prim> • <Edge> • <Edge#> Via edges: • <Et> • <Pse> • <Sel> • <Via> • <Layer> • <Layer_ID>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Sx> • <Sy> • <Ex> • <Ey> • <H> • <Rad>
	<*Port*>	String	Name of the selected port or pin.
	<*Et*>	String	Specifies the edge type: Pe for primitive edge or Pse for via edge.
	<*Prim*>	String	Specifies the primitive type associated with the edge.
	<*Edge#*>	Integer	Index number of the selected primitive.
	<*Sel*>	Integer	Selection index for the via. Used to identify which via instance is associated with the edge. Typically set to 0 unless multiple vias are present.
	<*Via*>	String	Name of the selected via.
	<*Layer*>	Integer	The ID of the layer of the pad of the via to use.
	<*Sx*>	Integer	X coordinate of the start point of the edge arc.
	<*Sy*>	Integer	Y coordinate of the start point of the edge arc.
	<*Ex*>	Integer	X coordinate of the end point of the edge arc.
	<*Ey*>	Integer	Y coordinate of the end point of the edge arc.
	<*H*>	Integer	Height of the edge arc (0 for a straight edge).
	<*Rad*>	Integer	Size of the edge arc in radians (0 for a straight edge).
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddRefPort([<port_list>], [<edge_array>])
Python Example	oEditor.AddRefPort(["port1", "port2"], ["NAME:Contents", "edge:=" , "edge_description", "edge:=" , "edge_description"])

AlignPorts (Layout Editor)

Causes automatic alignment of the microwave ports of the components, edge ports, and pins referenced in the argument. If the list is empty, aligns all the microwave ports in the layout.

UI Access	From **Draw**, select **Align MW Ports**.
Parameters	See table below
Return Value	None

Name	Type	Description
<*align_parameters*>	Array	Structured array containing these parameters in order: • <NAME:elements> • <object_name>
<*NAME:elements*>	String	Required first parameter which must have the specified value (i.e., "elements").
<*port_name*>	String	A comma-separated list of port names (e.g., "port1", "port2", "port3", et cetera).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AlignPorts(<*NAME:elements*>, <*port_name*>)

Python Example	`oEditor.AlignPorts(`
	`  ["NAME:Elements",`
	`  "port1",`
	`  "port2"`
	`  ])`

AssignCircuitRefPort (Layout Editor)

Assigns the internal port as a reference to an existing port (i.e., model as a circuit port).

UI Access	None		
Parameters	Name	Type	Description
	<*Port List*>	Array	Structured array containing a comma-separated list of any number of <Port_Name> parameters (e.g., "port1", "port2", "port3", et cetera).
	<*Via*>	String	Name of the selected via.
	<*Layer*>	Integer	The ID of the layer of the pad of the via to use.
	<*Sx*>	Integer	X coordinate of the start point of the edge arc.
	<*Sy*>	Integer	Y coordinate of the start point of the edge arc.
	<*Ex*>	Integer	X coordinate of the end point of the edge arc.
	<*Ey*>	Integer	Y coordinate of the end point of the edge arc.
	<*H*>	Integer	Height of the edge arc (0 for a straight edge).
	<*Rad*>	Integer	Size of the edge arc in radians (0 for a straight edge).
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddRefPort([<port_list>], [<edge_array>])
Python Example	oEditor.AddRefPort(["port1", "port2"], ["NAME:Contents", "edge:=", ["edge_description"], "edge:=", ["edge_description"]])

ClearRefPort (Layout Editor)

Clears reference ports from each of the specified named ports.

UI Access	From **Draw** > **Clear Reference Port**.		
Parameters	Name	Type	Description
	<PortList>	Array	Array of port names from which reference ports will be cleared.
Return Value	None		

Python Syntax	ClearRefPort(<PortList>)
Python Example	ClearRefPort(["Port1", "Port2", "Port3"])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateCircuitPort (Layout Editor)

Creates a circuit port between two points.

UI Access	From **Draw**, select **Create Circuit Ports**.
Parameters	

Name	Type	Description
<*NAME:Location*>	String	Object descriptor header.
<*PosLayer*>	String	Positive terminal layer name.
<*X0*>	Integer	X coordinate of the positive terminal.
<*Y0*>	Integer	Y coordinate of the positive terminal.
<*NegLayer*>	String	Negative terminal layer name.
<*X1*>	Integer	X coordinate of the negative terminal.
<*Y1*>	Integer	Y coordinate of the negative terminal.

Return Value	None
Python Syntax	CreateCircuitPort([<*NAME:Location*>, <*PosLayer*>, <*X0*>, <*Y0*>, <*NegLayer*>, <*X1*>, <*Y1*>])
Python Example	

```
oEditor.CreateCircuitPort(
  [
       "NAME:Location",
       "PosLayer:=", "TopLayer",
       "X0:=", "5mm",
       "Y0:=", "10mm",
       "NegLayer:=", "BottomLayer",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "X1:=", "5mm",
        "Y1:=", "0mm"
      ]
)
```

CreateEdgePort (Layout Editor)

Creates an edge port using the specified edges.

UI Access	**Draw** > **Port** > **Create** or **Right-click** > **Port** > **Create**.

Parameters

Name	Type	Description
<*EdgePortArray*>	Array	Identifies the contents of the array (i.e., "Contents") containing the following arguments: <NAME:Contents> <Edge> <External>
<*NAME:Contents*>	String	Identifies the contents of the array (i.e., "Contents").
<*Edge*>	Array	Array containing separate arrays for primitive edges and via edges. For primitive edges, array contains: <Et> <Prim> <Edge> For via edges, array contains:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		<Et> <Sel> <Layers> <Sx> <Sy> <Ex> <Ey> <H> <Rad> **Note:** Array can repeat in the same command.
<*Et*>	String	Type of new edge port (i.e., "pe" for primitive edge or "pse" for pad-stack edge or via).
<*Prim*>	String	Name of the target primitive.
<Edge#>	Integer	Edge number of the target primitive.
<Sel>	String	Name of the target via.
<Layers>	Integer	Layer ID of the pad of the target via.
<Sx>	Integer	Starting X coordinate for the arc of the new edge port.
<Sy>	Integer	Starting Y coordinate for the arc of the new edge port.
<Ex>	Integer	Ending X coordinate for the arc of the new edge port.
<Ey>	Integer	Ending Y coordinate for the arc of the new edge port.
<H>	Integer	Arc height of the new edge port (e.g., enter 0 for a straight edge).
<*rad*>	String	Arc size of the new edge port (e.g., enter 0 for a straight edge).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<External>	Boolean	True — the port is external. False — the port is internal.
Return Value	Text containing the name of the created edge port. (Returns an empty name if the edge port is not created.)		

Python Syntax	CreateEdgePort(<*EdgePortArray*>)

Python Example	Primitive edge port:
	`oEditor.CreateEdgePort (("NAME:Contents", "edge:=", ("et:=", "pe",`
	`"prim:=", "rect_167", "edge:=", 0), "edge:=", ("et:=", "pe", "prim:=",`
	`"rect_167", "edge:=", 3), "external:=", true))`
	Via edge port:
	`oEditor.CreateEdgePort (("NAME:Contents", "edge:=", ("et:=", "pse",`
	`"sel:=", "via_0", "layer:=", 10, "sx:=", -0.0015, "sy:=", -0.0015, "ex:=", 0.0015,`
	`"ey:=", -0.0015, "h:=", 0, "rad:=", 0), "external:=", true))`

CreateNPort (Layout Editor)

Creates an N-Port definition and component and adds them to the current project, layout, and schematic.

UI Access	None.		
Parameters	Name	Type	Description
	<*NPort_Array*>	Array	An array consisting of the following parameters in order: • <*NAME:Contents*>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• *<definition_name>* • *<placement>* • *<layer>*
	<NAME:Contents>	String	Identifies the contents of the N-Port definition array.
	<definition_name>	String	Name of the N-Port component definition.
	<placement>	String	Placement array containing parameters in order: "x:=", <value> — X coordinate; "y:=", <value> — Y coordinate; "scaling:=", <value> — Scaling factor (double).
	<layer>	String	Name of the layer on which the component is placed and the N-Port data definition, separated by commas.
Return Value	Component name		

Python Syntax	oEditor.CreateNPort([<NAME:Contents>, <definition_name>, <placement>, <layer>])

Python Example	`oEditor.CreateNPort(`
	`  [`
	`  "NAME:Contents",`
	`  "definition_name:=", "MyNPort",`
	`  "placement:=", ["x:=", 0.0, "y:=", 0.0, "scaling:=", 1.0],`
	`  "layer:=", "TopLayer"`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreatePortInstancePorts (Layout Editor)

Use: Create port on port instances.

Command: Right-Click-Menu > Port > Create

Syntax: CreatePortInstancePorts (["NAME:elements", "element-name",…])

Return Value: None

Parameters: <element-name>

Type: string

// Name of a port instance on which ports will be created.

CreatePortsOnComponents (Layout Editor)

Creates port-on-port instances of selected components.

UI Access	Right click on component(s) to select **Port** > **Create Ports on Component**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Name*></td><td>String</td><td>"NAME:elements"</td></tr><tr><td><*element-name*></td><td>String</td><td>Name of a component</td></tr></table>
Return Value	None. Ports are created on the unconnected port instances of the selected components.

Python Syntax	CreatePortsOnComponents(["NAME:elements", <element-name>,…])

Python Example	oEditor.CreatePortsOnComponents(
	  [
	  "NAME:elements",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "poly101",
    "rect100"
  ])
```

CreatePortsOnComponentsByNet (Layout Editor)

UI Access	After selecting one or more components, right-click within the **Layout Editor** and select **Port** > **Create Ports On Component** to open the **Create Ports** window. Select a net from the **Net Filter** area, then click **OK** to close the **Create Ports** window.
Parameters	

Name	Type	Description
<Parameters>	Array	Structured array containing these parameters in order: • <NAME:Components> • <ComponentNames>
<NAME:Components>	String	Required first parameter which must have the specified value (i.e., "Components").
<ComponentNames>	String	Names of the target components, separated by commas.
<Parameters>	Array	Structured array containing these parameters in order: • <NAME:Nets> • <NetNames>
<NAME:Nets>	String	Required first parameter which must have the specified value (i.e., "Nets").
<NetNames>	String	Names of the target net(s), separated by commas (if more than one).
<Type>	String	Required parameter which must have the value "Port".
<R>	Integer	Value of resistor.
<L>	Integer	Value of inductor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*C*>	Integer	Value of capacitor.
Return Value	None.		

Python Syntax	CreatePortsOnComponentsByNet([<NAME:Components>, <ComponentNames>], [<NAME:Nets>, <NetNames>], <Type>, <R>, <L>, <C>)

Python Example	`oEditor.CreatePortsOnComponentsByNet(`
	`  [`
	`  "NAME:Components",`
	`  "U0"`
	`  "U1"`
	`  ],`
	`  [`
	`  "NAME:Nets",`
	`  "trace"`
	`  ], "Port", "0", "0", "0")`

GetPortInfo (Layout Editor)

Use: Request information on a port or pin.

Command: None.

Syntax: GetPortInfo<"name">

Return Value: an array of text as follows:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

For pins and ports that are not edge ports:

Name=<name>

Type=Pin, Padstack: <padstack definition name>

X=<X coordinate>

Y=<Y coordinate>

ConnectionPoints=<connection description>; <connection description>,...

NetName=<net name>

For edge ports:

Name=<name>

Type=EdgePort

X=<X coordinate>

Y=<Y coordinate>

ConnectionPoints=<connection description>; <connection description>,...

NetName=<net name>

<name>

Text containing the name of the pin or port

<padstack definition name>

Text containing the name of the associated padstack definition

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<X coordinate>

Double indicating the X location of the pin or port location

<Y coordinate>

Double indicating the Y location of the pin or port location

<connection description>

< X coordinate> <Y coordinate> Dir:<direction> Layer: <layer name>

<direction>

Either NONE or a double giving the angle in degrees for the connection direction

<layer name>

Name of the layer for the connection point being described.

Example returned values:

Name=Pin1

Type=Pin, Padstack: Padstack

X=0.000744

Y=0.015537

ConnectionPoints= 0.000744 0.015537 Dir:NONE Layer: Top; 0.000744 0.015537 Dir:NONE Layer: Ground; 0.000744 0.015537 Dir:NONE Layer: Bottom

NetName=Pin1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Name=Port1

Type=EdgePort

X=-0.008120

Y=0.004264

ConnectionPoints= -0.008120 0.004264 Dir:90.000000 Layer: Bottom

NetName=Port1

Parameters: <"name">

Text that contains the name of the port or pin for which information is being requested.

RemovePort (Symbol Editor)

Selected pins are changed to vias. Selected edge terminal ports are removed.

UI Access	**Not Available**		
Parameters	Name	Type	Description
	<*Name*>	String	"NAME:elements"
	<*ObjectName*>	String	Name of object to be removed. Multiple objects can be listed, each as a separate string
Return Value	None		

Python Syntax	RemovePort ([<NAME:elements", <*ObjectName*> ...])
Python Example	`oEditor.RemovePort(["NAME:elements", "via_195", "Port1"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemovePortsFromAllNets (Layout Editor)

Removes ports from all the pins in all the nets.

UI Access	Layout tab under Nets right-click and click **Remove Ports**.
Parameters	None
Return Value	None

Python Syntax	RemovePortsFromAllNets
Python Example	`oEditor.RemovePortsFromAllNets()`

RemovePortsFromNet (Layout Editor)

Removes ports from all the pins on the designated nets.

UI Access	In the **Layout** window, right click the net and click **Port** > **Remove Ports from Net** or on the **Nets** tab of the **Nets** window, highlight and right click one or more nets and choose **Remove Ports**.
Parameters	Name: <*requiredKeyword*> — Type: String — Description: Required parameter that must proceed all element names with the value "NAME:Nets". Name: <*netName*> — Type: String — Description: Name of a net. Ports are removed from the port instances of the selected nets.
Return Value	None.

Python Syntax	RemovePortsFromNet(["NAME:Nets", <*netName*>, ...])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.RemovePortsFromNet(`
	`  [`
	`  "NAME:Nets",`
	`  "NET_1",`
	`  "NET_2",`
	`  "NET_3"`
	`  ])`

RemovePortsOnComponents (Layout Editor)

Remove ports on port instances of selected components. Multiple nets can be listed.

UI Access	Right click and choose **Port** > **Remove Ports From Component**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*requiredKeyword*></td><td>String</td><td>Required parameter that must proceed all element names with the value "NAME:elements".</td></tr><tr><td><*elementName*></td><td>String</td><td>Name of a component. Ports are removed from the port instances of the selected components.</td></tr></table>
Return Value	None.

Python Syntax	RemovePortsOnComponents (["NAME:elements", <*elementName*>,...])
Python Example	`oEditor.RemovePortsOnComponents(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:elements", "0"])

SetHfssExtentsVisible (Layout Editor)

Hides or reveals HFSS extents.

UI Access	From the **Layout** tab, select **HFSS Extents** > **Show**/**Hide**.
Parameters	**Name** / **Type** / **Description** *<Enable>* / Boolean / **True** displays HFSS extents. **False** hides HFSS extents.
Return Value	None.

Python Syntax	SetHfssExtentsVisible(<Enable>)
Python Example	`oEditor.SetHfssExtentsVisible(False)`

SetHfssPortsVisible (Layout Editor)

Enables or disables target components.

UI Access	From **Layout**, select **Draw Ports/Sources**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	**Name**	**Type**	**Description**
	<Enable>	Boolean	**True** displays HFSS ports. **False** hides HFSS ports.
Return Value	None.		

Python Syntax	SetHfssPortsVisible(<Enable>)
Python Example	`oEditor.SetHfssPortsVisible(False)`

Miscellaneous Methods

The following miscellaneous methods are available.

AlignObjects (Layout Editor)

ChangeOptions (Layout Editor)

ChangeProperty (Schematic Editor and Layout Editor)

ClipPlane (Layout Editor)

CopyToHFSS 3D Layout (Layout Editor)

CutOutSubDesign (Layout Editor)

DefeatureObjects (Layout Editor)

DissolveComponents (Layout Editor)

Duplicate (Layout Editor)

Edit3DComponentDefinition (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditComponent (Layout Editor)

EditPinGroup (Layout Editor)

EnableComponents (Layout Editor)

EraseMeasurements (Layout Editor)

ExportDXF (Layout Editor)

ExportGDSII (Layout Editor)

ExportGerber (Layout Editor)

ExportNCDrill (Layout Editor)

GetActiveUnits (Layout Editor)

GetComponentInfo (Layout Editor)

GetCompInstanceFromRefDes (Layout Editor)

GetComponentPins (Layout Editor)

GetComponentPinInfo (Layout Editor)

GetEditorName (Layout Editor)

GetMaterialList (Layout Editor)

GetProperties (Layout Editor)

GetPropertyValue (Layout Editor)

GetSelections (Layout Editor)

Heal (Layout Editor)

PageSetup (Layout Editor)

PushExcitations (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Select (Layout Editor)

SelectAll (Layout Editor)

SetActiveUnits (Layout Editor)

SetPropertyValue (Layout Editor)

StitchLines (Layout Editor)

UnselectAll (Layout Editor)

ZoomToFit (Layout Editor)

AlignObjects (Layout Editor)

Aligns objects (e.g. primitives, components, et cetera) relative to each other.

<table>
<tr><td>UI Access</td><td colspan="3">From Draw, select Align.</td></tr>
<tr><td rowspan="4">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><align_ parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:align>
<horz_align>
<vert_alignment</td></tr>
<tr><td><NAME:align></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "align").</td></tr>
<tr><td><horz_align></td><td>String</td><td>Optional. Must be any one of the following:
Leave Blank — no horizontal alignment.
AlignLeft — align objects left.
AlignRight — align objects right.
CenterHorz — align object centers horizontally.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• DistributeCentersHorz — places objects so the horizontal spaces between their centers are equal . • SpaceEdgesHorz — places objects so the horizontal spaces between their edges are equal
	<*vert_align*>	String	***Optional.*** Must be any one of the following: • Leave Blank — no vertical alignment. • AlignTop — align tops of objects. • AlignBottom — align bottoms of objects. • CenterVert — align object centers vertically. • DistributeCentersVert — places objects so the vertical spaces between their centers are equal . • SpaceEdgesVert — places objects so the vertical spaces between their edges are equal .
	<*Element_Parameters*>	Array	Structured array containing these parameters in order: • <NAME:elements> • <elements>
	<*NAME:elements*>	String	Required first parameter which must have the specified value (i.e., "elements").
	<*elements*>	String	A comma-separated list of object names (e.g., "polygon_01", "polygon_02", "polygon_03", et cetera).
Return Value	None		

Python Syntax	AlignObjects([<*NAME:align*>, <*horz_align*>, <*vert_align*>], [<NAME:elements>, <elements>])
Python Example	`oEditor.AlignObjects(` `  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NAME:align",
    "AlignLeft",
    "AlignTop"
  ],
  [
    "NAME:elements",
    "polygon_01",
    "polygon_02"
  ])
```

of y

ChangeOptions (Layout Editor)

Changes options for an existing layout. Does not change global options specified in the registry. Only those options being changed need to be specified. Options not specified are not affected.

UI Access	Access via scripting only. No direct UI command.

Parameters

Name	Type	Description
"NAME:Options"	String	Required header for the options array. Must be the first item in the list.
<OptionData>	Mixed	Option name and value pair. Value can be: **String** (e.g., `"SnapMode:=", "Fine"`) **Float** (e.g., `"DrawingResolution:=", 64.0`)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Array of floats (e.g., `"PageExtent:=", [-0.1, -0.1, 0.1, 0.1]`)
Return Value	None		

Python Syntax	oEditor.ChangeOptions["NAME:options", <OptionData>, …])

Python Example	`oEditor.ChangeOptions([`
	`  "NAME:Options",`
	`  "useFixedDrawingResolution:=", False,`
	`  "DrawingResolution:=", 64,`
	`  "ShowGrid:=", True,`
	`  "MajorSize:=", 1.0,`
	`  "MinorSize:=", 0.1,`
	`  "PageExtent:=", [-0.1, -0.1, 0.1, 0.1],`
	`  "DefaultToSketchMode:=", False,`
	`  "NetColor:=", 16777152`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

An error will be returned if this script command is not used at the top-level hierarchy.

- Global layout defaults are stored in the registry (Layout\Preferences\)
- Names of registry items were selected for backwards compatibility and are consistent with adsn, technology file, and scripting naming
- Local options are both script-able and undo-able
- Global options are neither script-able nor undo-able

Option	Description	Installed default	Registry, Scripting Names, Data Type
Arc Drawing Resolution	Controls drawing of segments for arcs - either dynamic and based on current zoom or fixed to specified value	Dynamic	"useFixedDrawingResolution"[1] "DrawingResolution" [2]
Major and Minor Grid lines	Spacing, color, and visibility	The grid is displayed. Spacing based on current default length units. Major grid lines are 10 minor grid lines apart. The line colors are light blue grays, with major grid lines being darker.	"MajorColor" [3] "MinorColor" [3] "MajorSize" [2] "MinorSize" [2] "ShowGrid" [1]
Drawing Extent	Specifies size and coordinates of the layout	Lower left of the layout is (-0.1, -0.1) and the upper right is (0.1, 0.1)	"PageExtent" [4]
Background	Color of the layout background	white	"background color" [3]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Color			
Sketch Mode	Fill patterns and center lines are not drawn.	off	"DefaultToSketchMode" [1]
Solid Mode	Objects are filled in with solid color.	off	"fillMode" [1]
Draw Connection Points	Display pin symbols in the layout.	off	"show connection points" [1]
Draw Rats	Display lines indicating missing physical connections in nets	on	"draw rats" [1]
Label Vertices	Display property text labeling the vertices of selected polygons and lines.	off	"display vertex labels" [1]
Net Color	Color used for highlighting nets	light yellow	"NetColor" [3]
Rectangle Description	Specifies if rectangles are described with two points or center point, width, and height	2 points	"rectangle description" [5]
Default Hole Size	Default size for actual holes in a PC board	25 mil	"defaultholesize" [2]
Align Microwave Components	Snaps components on entry.	on	"snaptoport" [1]
Symbol Footprint Scaling	Used to size symbol footprints placed in layout.	Based on data extent and current length units.	"sym footprint scaling" [6]
Primary and Secondary Selection Colors	Colors used to indicate the first item selected and other items currently selected.	Primary color is red, secondary color is a darker red	"primary selectioncolor" [3] "secondary selection color" [3]
Preview Selection	Highlight the object that would be selected with a left mouse button	off	"preview selection" [1]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	click in the current location.		
Snapping Options	Choose snapping sources and targets and the maximum distance (in pixels) for snapping to occur. Constrain edits to grid if desired. **Note:** Off-grid objects are ignored if ConstrainToGrid is asserted.	Snap distance is 20 pixels. Snapping sources are vertex, edge center, object center, & elec. conn. Snapping targets are vertex, edge center, object center, elec. conn, & grid. Edits are constrained to grid.	"PixelSnapTolerance"[7] "SnapTargetVertex_on"[1] "SnapTargetEdgeCenter_on"[1] "SnapTargetObjCenter_on"[1] "SnapTargetEdge_on"[1] "SnapTargetElecConnection_on"[1] "SnapTargetIntersection_on"[1] "SnapTargetGrid_on"[1] "SnapSourceVertex_on"[1] "SnapSourceEdgeCenter_on"[1] "SnapSourceObjCenter_on"[1] "SnapSourceEdge_on"[1] "SnapSourceElecConnection_on"[1] "ConstrainToGrid"[1]
Always Show Merge Layers Dialog	Controls display of the layer merging dialog.	off – Only show dialog when the user must be involved.	"AlwaysShowLayerMergeDlg"[1]
Rotation Increment	During rotation, objects are rotated by multiples of this amount.	Based on current angle units.	"anglesnap" [2]
Allow Drag on First Click	Selection and move with one left mouse button click.	false	"AllowDragOnFirstClick" [1]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

[1] Boolean

[2] Integer

[3] String

[4] [float, float, float, float]

[5] Enumeration

ChangeProperty (Schematic Editor and Layout Editor)

Changes properties using the ChangeProperty command. This command can be executed by oEditor to change editor properties, by oDesign to change design-level properties, and by oProject to change project-level properties. It supports creation, editing, and deletion of properties. In HFSS and Maxwell, only Variable and Separator properties can be deleted.

Note: The `ChangeProperty` command supports multiple tab targets such as `DesignTab`, `ComponentTab`, and others. Property names and fields vary depending on the tab context.

Use the script recording feature to edit a property and view the resulting script entry, or use GetPropertyValue to inspect the expected format.

<table>
<tr><td>UI Access</td><td colspan="3">None</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PropTabArray></td><td>Array</td><td>Structured array containing tab-level property changes. See detailed structure below.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ChangeProperty(["Name:AllTabs", <PropTabArray>, <PropTabArray>, ...])

Python Example	`oEditor.ChangeProperty([`
	`  "NAME:AllTabs",`
	`  ...PropTabArray entries...`
	`  ])`

Property Array Structure

The following structure defines the nested arrays and parameters used in `ChangeProperty`. Each array must follow the specified order and naming convention.

```
  ["Name:<PropTab>",
    <PropServersArray>,
    <NewPropsArray>,
    <ChangedPropsArray>,
    <DeletedPropsArray>]
<PropServersArray>
  ["Name:PropServers", <PropServer>, <PropServer>, ...]
<NewPropsArray>
  ["Name:NewProps", <PropDataArray>, <PropDataArray>, ...]
<ChangedPropsArray>
  ["Name:ChangedProps", <PropDataArray>, <PropDataArray>, ...]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<DeletedPropsArray>
   ["Name:DeletedProps", <PropName>, <PropName>, ...]
<PropDataArray>
   ["NAME:<PropName>",
     "PropType:=", <PropType>,
     "NewName:=", <string>,
     "Description:=", <string>,
     "NewRowPosition:=", <int>,
     "ReadOnly:=", <bool>,
     "Hidden:=", <bool>,
     <PropTypeSpecificArgs>]
<PropType>
```

Type: string

Identifies the type of property when a new property is added. In HFSS and Maxwell, only separator properties and variable properties can be added.

```
   "SeparatorProp"
   "VariableProp"
   "TextProp"
   "NumberProp"
   "ValueProp"
   "CheckboxProp"
   "MenuProp"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"PointProp"

"VPointProp"

"V3DPointProp"

"ButtonProp"

NewName

Specify the new name of a property if the property's name is being edited. In HFSS and Maxwell, the name can only be changed for separators and variables.

Description

Specify a description of the property. In HFSS and Maxwell, the description can only be changed for separators and variables.

NewRowPosition

Used to reorder rows in the **Property** dialog box.

In HFSS, this only applies to the **Project** > **Project Variables** panel and the **HFSS** > **DesignProperties** panel.

In Maxwell, this only applies to the **Project** > **Project Variables** panel and the **Maxwell3D** > **DesignProperties**, **Maxwell2D** > **Design Properties**, or **RMxprt** > **Design Properties** panels. Specify the new zero-based row index of the variable or separator.

ReadOnly

Used to mark a property as "read only" so it cannot be modified. In HFSS and Maxwell, this flag can only be set for variables and separators.

Hidden

Used to hide a property so it cannot be viewed outside of the **Property** dialog box. In HFSS and Maxwell, this flag can only be set for variables and separators.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<PropTypeSpecificArgs>
  SeparatorProp: no arguments
  TextProp: "Value:=", <string>
  NumberProp: "Value:=", <double>
  ValueProp: "Value:=", <value>
  CheckboxProp: "Value:=", <bool>
  MenuProp: "Value:=", <string>
  PointProp: "X:=", <double>, "Y:=", <double>
  VPointProp: "X:=", <value>, "Y:=", <value>
  V3DPointProp: "X:=", <value>, "Y:=", <value>, "Z:=", <value>
  Material Button: "Material:=", <string>
  Color Button: "R:=", <int>, "G:=", <int>, "B:=", <int>
  Transparency Button: "Value:=", <double>
```

`<PropTypeSpecificArgs>` for VariableProps

Syntax:

```
"Value:=", <value>, <OptimizationFlagsArray>, <TuningFlagsArray>, <SensitivityFlagsArray>,
<StatisticsFlagsArray>
```

Parameters:

```
<OptimizationFlagsArray>
  ["NAME:Optimization",
  "Included:=", <bool>,
  "Min:=", <value>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Max:=", <value>]
  <TuningFlagsArray>
    ["NAME:Tuning",
    "Included:=", <bool>,
    "Step:=", <value>,
    "Min:=", <value>,
    "Max:=", <value>]
  <SensitivityFlagsArray>
    ["NAME:Sensitivity",
    "Included:=", <bool>,
    "Min:=", <value>,
    "Max:=", <value>,
    "IDisp:=", <value>]
  <StatisticsFlagsArray>
    ["NAME:Statistical",
    "Included:=", <bool>,
    "Dist:=", <Distribution>,
    "StdD:=", <value>,
    "Min:=", <value>,
    "Max:=", <value>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Tol:=", <string>]
```

`<Distribution>`

Type: string

Value should be "Gaussian" or "Uniform"

`StdD`

Standard deviation.

`Min`

Low cut-off for the distribution.

`Max`

High cut-off for the distribution.

`Tol`

Tolerance for uniform distributions. Format is "<int>%". Example: "20%".

ClipPlane (Layout Editor)

Truncates graphics at the XY-plane in the +Z direction. Clip planes can be used to truncate graphics in the Layout view, including mesh and fields plots.

UI Access	From the **View** menu, select **Clip Plane**, or from the **View** ribbon, select **Add Clipping Plane**.
Parameters	None.
Return Value	None.

Python Syntax	ClipPlane()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	oEditor = oDesign.SetActiveEditor("Layout") Editor.ClipPlane ("*n*1, *n*2, *n*3, et cetera") success = oEditor.ClipPlane("*n*1, *n*2, *n*3, et cetera")

CopyToHFSS 3D Layout (Layout Editor)

Use: Generate a single, flat, HFSS 3D Layout design from a selection of components and/or sub-circuits.

Command: CopyToHFSS 3D Layout

Syntax: CopyToHFSS 3D Layout Array("NAME:elements", <"obj1">, <"obj2">, …)

Return Value: None

Parameters: <"obj1">

Type: text

// component or sub-circuit instance name.

CutOutSubDesign (Layout Editor)

Use: Cut out a subdesign.

Command: CutOutSubDesign

Syntax: oEditor.CutOutSubDesign Array(

"NAME:Params",

"Name:=", <"name">,

"EMDesign:=", <boolean>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"SubDesign:=", <boolean>,

"Within:=", <boolean>,

"Without:=", <boolean>,

"AutoGenExtent":=<boolean>,

"Expansion":=<double>,

"RoundCorner":=<boolean>,

"Increments":=<integer>,

"ExtentSel:=", Array(<"extent-poly">, …),

Array("NAME:Nets", "net:=", Array(<"net-name">, <clip>), …))

In place of "ExtentSel:=", Array(<"extent-poly">, …) an explicit polygon can be used:

"Extent:=", Array("cl:=", true,

"pt:=", Array(U:=", <"units">, "x:=", <double>, "y:=" <double>, …))

Parameters: Name — name of the cutout design.

EMDesign — if true, create an EM design, otherwise create a Nexxim design.

Within — boolean; cut out the interior region.

Without — boolean; cut out the exterior region.

AutoGenExtent — boolean; when true, Circuit disregards both ExtentSel and Extent. Circuit will instead generate a polygonal shape around all nets which are not clipped.

Expansion — double; similar to the Auto Generate Extent dialog box. However, this is always a unitless fraction. By default it is set to .1.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RoundCorner — boolean; identical to the Auto Generate dialog box. By default this is set to true. In general, rounded corners are preferred when the cutout shape has acute or nea-acute angles

Increments — integer; this is from the Auto Generate Dialog, and by default is set to true. This can greatly increase the running time and a small increase can make a big difference. It is probably best to experiment with this parameter first, rather than set it arbitrarily.

ExtentSel— an array of extent polygon names.

Extent — alternative to ExtentSel; an explicit polygon defined by coordinates.

cl — must always be true; indicates that the polygon is closed.

U — coordinate units, e.g. "mm"

pt — array of coordinate values.

x — x coordinate value

y — y coordinate value

Nets — the net information.

net — array of net information.

<"net-name">

"<no net trace>" — special value referring to traces not in a net.

"<no net fill>" — special value refereeing to fill polygons not in a net.

<design>:<net> — net within a particular design.

<net> — net within the active design.

<clip> — boolean true/false; if true, the net is clipped against the extent else the net is included but not clipped.

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: Using "extent_poly" as the selection extent:

```
oEditor.CutOutSubDesign Array("NAME:Params",
"Name:=", "EMDesign1_cutout",
"EMDesign:=", true,
"SubDesign:=", false,
"Within:=", true,
"Without:=", false,
"ExtentSel:=", Array("extent_poly", …),
Array("NAME:Nets",
"net:=", Array("<no net trace>", true),
"net:=", Array("<no net fill>", true),
"net:=", Array("EMDesign1:GND", true) … ))
```

Example: Example using an explicit polygon as the selection extent:

```
oEditor.CutOutSubDesign Array("NAME:Params",
"Name:=", "EMDesign1_cutout",
"EMDesign:=", true,
"SubDesign:=", false,
"Within:=", true,
"Without:=", false,
"Extent:=", Array(
"cl:=", true,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"pt:=", Array(
"U:=", "mm",
"x:=", 0,
"y:=" 0, ) ),
Array("NAME:Nets",
"net:=", Array("<no net trace>", true),
"net:=", Array("<no net fill>", true),
"net:=", Array("EMDesign1:GND", true) … ))
```

Defeature Objects (Layout Editor)

Use: Remove irrelevant features from a primitive. Vertices that deviate less than the tolerance from a chord are removed; narrow concave regions less than the tolerance wide are also eliminated. The command can also be used to fix self-intersections (only the 'positive' areas are kept).

Command: DefeatureObjects

Syntax: DefeatureObjects Array("NAME:options", "tolerance:=", <value>, "fix:=", <fix>), Array("NAME:elements", <"primitive">, …)

Return Value: None

Parameters: <"value">

Type: double, e.g. 0.001

Description: tolerance value

<fix>

Type: boolean (true or false)

Description: if true, then self-intersections are fixed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

< primitive >

Type: text

Description: primitive name.

DissolveComponents (Layout Editor)

Dissolves the named components.

UI Access	After selecting one or more vias, right-click within the Layout Editor and select **Components** > **Dissolve**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array containing these parameters in order: <NAME:elements> <TargetObjects> </td></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "vias").</td></tr><tr><td><*TargetObjects*></td><td>String</td><td>Names of the target primitives, vias, or groups to dissolve, separated by commas.</td></tr></table>
Return Value	None.

Python Syntax	DissolveComponents([<NAME:elements>, <TargetObjects>])

Python Example	`oEditor.DissolveComponents(`
	`  ["NAME:elements",`
	`  "poly_801",`
	`  "ViaGroup_2"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Duplicate (Layout Editor)

Use: Duplicates layout objects.

Command: The specified objects are duplicated by the given count with the specified offset.

Syntax: Duplicate Array("NAME:options", "count:=", <count data>), Array("NAME:elements", <element names>), Array(<x>, <y>)

Return Value: None

Parameters: <count data> is the number of sets of new objects to be generated (and does not include the original objects)

<element names> is one or more strings with the names of layout objects

<x> is the x value for the offset

<y> is the y value for the offset

Edit3DComponentDefinition

Edits definitions of a specified 3D component.

<table>
<tr><td>UI Access</td><td>Layout > 3D Component Definitions. Select component and click Edit Definition.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:3D Components",</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Definitions:=" , [<string component name>], "IsLocal:=" , <boolean True for local definition; False to save to file>, "EditFilePath:=", <string path of new component file location>] If saving locally or leaving the file path unaltered, pass empty string for EditFilePath.
Return Value	None.		
Python Syntax	Edit3DComponentDefinition (<*Parameters*>)		
Python Example	oEditor.Update3DComponentDefinitions(["NAME:3D Components", "Definitions:=" , ["My_Component"], "IsLocal:=" , True, "EditFilePath:=", "")		

EditComponent (Layout Editor)

Modifies a current source.

idt the selected component to add or remove pins

UI Access	After choosing a component, navigate to the**Properties** window. From the **Pins** row, select **Edit** to open the **Edit Component** window. From the **Edit Component** window, remove pins as necessary.		
Parameters	Name	Type	Description
	<*Parameters*>	Array	Structured array containing these parameters in order:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <NAME:Contents> • <definition_name> • <ref_des:=> • <add> • <modified_elements>
	<NAME:Contents>	String	Required first parameter which must have the specified value (i.e., "Contents").
	<definition_name>	String	Identifies the component (e.g., Component_0).
	<ref_des>	String	Identifies the selected component's reference port (e.g., U0).
	<add>	Boolean	**True** adds pins **False** removes pins
	<modified_elements>	Array	Structured array containing the names of pins to add or remove (e.g., via_246, via_247).
Return Value	None.		

Python Syntax	EditComponent([<NAME:Contents>, <definition_name>, <ref_des>, <add>, <modified_elements>])

Python Example	`oEditor.EditComponent(`
	`[`
	`  "NAME:Contents",`
	`  "definition_name:=", "Component_0",`
	`  "ref_des:=", "U0",`
	`  "add:=", "False",`
	`  "modified_elements", ["via_246", "via_247"]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

EditPinGroup (Layout Editor)

Allows the user to edit the target pin group.

UI Access	From **Draw**, select **Create/Manage Pin Groups** to open the **Create/Manage Pin Groups** window. Select a group from the **Pin Group List**, then click **Edit Pin Group** to open the **Edit Pin Group** window. Check the adjacent box next to any pins to remove them, then click **Delete Pins**. Click **OK** to close the **Edit Pin Group** window.
Parameters	See table below
Return Value	None.

Name	Type	Description
<*NAME*>	String	Name of the target pin group.
<*Parameters1*>	Array	Structured array containing these parameters in order: • <NAME:elements_remove> • <PinsRemove>
<*NAME:elements_remove*>	String	Required first parameter which must have the specified value (i.e., "elements_remove").
<*PinsRemove*>	String	Names of the pins to remove from the group, separated by commas.
<*Parameters2*>	Array	Structured array containing these parameters in order: • <NAME:elements_add> • <PinsAdd>
<*NAME:elements_add*>	String	Required first parameter which must have the specified value (i.e., "elements_add").
<*PinsAdd*>	String	Names of the pins to add to the group, separated by commas.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditPinGroup(<*NAME*>, [<NAME:elements_remove>, <PinsRemove>], [<NAME:elements_add>, <PinsAdd>])

Python Example	`oEditor.EditPinGroup("GND_30_SQFP28X28_208_U1",`
	`["NAME:elements_remove",`
	`"U1-50",`
	`"U1-82"],`
	`["Name:elements_add",`
	`"U1-51"`
	`])`

EnableComponents (Layout Editor)

Enables or disables target components.

UI Access	From the **Components** window, right-click a component and select **Enable**/**Disable** or **Ctrl**+click multiple components, then right-click any of the selected components and click **Enable**/**Disable**.

Parameters

Name	Type	Description
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:Components> • <ObjectName>
<*NAME:Components*>	String	Required first parameter which must have the specified value (i.e., "Components").
<*ObjectName*>	String	List of target objects, separated by commas.
<*Enable*>	Boolean	**True** enables selected components. **False** disables selected components

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	EnableComponents ([<NAME:Components>, <ObjectName>], (<Enable>)

Python Example	`oEditor.EnableComponents(`
	`  [`
	`  "NAME:Components",`
	`  "U0",`
	`  "U1"`
	`  ], False)`

EraseMeasurements (Layout Editor)

Use: Causes erasing of ALL measurements currently present.

Command: None.

Syntax: EraseMeasurements

Return Value: None.

Parameters: None.

ExportDXF (Layout Editor)

Use: Export DXF

Command: File > Export > AutoCAD

Syntax: ExportDXF

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Array("NAME:options", "FileName:=", <"filename">, "AcadVersion:=,<version>, "ScaleFactor:=", <scale>),

Array("NAME:layers", <"layer">, …)

Return Value: None

Parameters:

<"filename">

Type: text

Description: file path for the export file

<version>

Type: integer

Description: version of AutoCAD. Acceptable values are:

- **0** corresponding to AutoCAD version 13
- **1** corresponding to AutoCAD version 14
- **2** corresponding to AutoCAD 2000-2003
- **3** corresponding to AutoCAD 2004-2006
- **4** corresponding to AutoCAD 2007-2009
- **5** corresponding to AutoCAD 2010-2012

<scale>

Type: double

Description: the scaling of the values written out, e.g. if the data is written out in "mm", the scaling factor will be 1000.

<"layer">

Type: text

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Description: names of the layers to be exported.

ExportGDSII (Layout Editor)

Exports results to GDSII.

<table>
<tr><th>UI Access</th><th>File > Export > GDSII.</th></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OptionParameters></td><td>Array</td><td>Structured array.
<pre>["NAME:options",
  "FileName:=", <string, GDSII export file name.>,
  "NumVertices:=", <integer, maximum number of vertices allowed in a polygon (specify zero if unlimited).>,
  "ArcTol:=", <double, arc tolerance (in meters); the value used to discretize arcs. Smaller the value the greater the number of edges exported. For 0.1mm use 0.0001.>,
  "LayerMap:=", [
  "entry:=", [
  "layer:=", <string, name of a layer to export.>,
  "id:=", <integer, GDSII layer ID to assign the exported layer.>,
  "include:=", <boolean, if true, the layer is exported.>]</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			]]

Return Value	None.

Python Syntax	ExportGDSII(<*OptionParameters*>)
Python Example	oEditor.ExportGDSII(["NAME:options", "FileName:=", "C:/Projects/ design.gds", "NumVertices:=", 0, "ArcTol:=", 2E-006, "LayerMap:=", ["entry:=",["layer:=", "Bottom", "id:=", 2, "include:=", true], "entry:=", ["layer:=", "Top", "id:=", 1, "include:=", true]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]]

ExportGerber (Layout Editor)

Use: Export the current design to Gerber files.

Command: File > Export > Gerber

Syntax: ExportGerber

Array("NAME:options",

"FileName:=", <"file name">,

"Units:=", <"units">,

"IntPlace:=", <integer places>,

"DecimalPlace:=", <decimal places>,

"Suppress:=", <"suppress">,

Array("NAME:GerberPages",

"<page>:=", Array(<"layer">, …), …))

Return Value: None

Parameters: <"file name">

Type: text

Description: prefix for the exported Gerber files; the actual files will be named <file name>_<page>.ger

<"units">

Type: text

Description: unit code, e.g. "in", "mm", etc.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<integer places>

Type: integer

Description: number of integer places to use when formatting the numerical output.

<decimal places>

Type: integer

Description: number of decimal places to use when formatting the numerical output.

<"suppress">

Type: text

Description: either "Leading Zeros", or "Trailing Zeros".

<page>

Type: integer

Description: Gerber page number

<"layer">

Type: text

Description: Layers to be exported to the specified Gerber page.

ExportNCDrill (Layout Editor)

Use: Export the vias to an NC Drill file

Command: File > Export > NCDrill

Syntax: ExportNCDrill

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Array("NAME:options",
"FileName:=", <"file name">,
"Units:=", <"units">,
"IntPlace:=", <integer places>,
"DecimalPlace:=", <decimal places>,
"SuppressLeadingZero:=", <suppress leading zeros>,
Array("NAME:NCDFileOptsVec",
Array("NAME:NCDrillOptions",
"Top:=", <"layer">,
"Bottom:=", <"layer">,
"Create:=", <create>), …))
```

Return Value: None

Parameters: <"file name">

Type: text

Description: prefix for the exported Gerber files; the actual files will be named <file name>_<page>.ger

<"units">

Type: text

Description: unit code, e.g. "in", "mm", etc.

<integer places>

Type: integer

Description: number of integer places to use when formatting the numerical output.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<decimal places>

Type: integer

Description: number of decimal places to use when formatting the numerical output.

<suppress leading zeros>

Type: boolean

Description: if true, then suppress leading zeros in the output file.

<layer>

Type: text

Description: Start/end layer names; each layer range is written to a separate NC drill file.

<create>

Type: boolean

Description: if true, create the corresponding NC drill file.

GetActiveUnits (Layout Editor)

Returns the current active unit of measurement.

UI Access	N/A.
Parameters	None.
Return Value	Current active unit of measurement (e.g., mm).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetActiveUnits()
Python Example	`oEditor.GetActiveUnits()` `'mm'`

GetComponentInfo (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetComponentInfo<compID>

Return Value: array of strings, as follows:

"ComponentName=*string"*

"PlacementLayer=*string"*

"LocationX=*number"*

"LocationY=*number"*

"BBoxLLx=*number"*

"BBoxLLy=*number"*

"BBoxURx=*number"*

"BBoxURy=*number"*

"Angle=*number" (in degrees)*

"Flip=true *or* false"

"Scale=*number"*

Parameters: The LayoutComp's ID

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetCompInstanceFromRefDes (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetCompInstanceFromRefDes (<object_name>)

Return Value: Returns IDispatch for CompInstance.

```
Parameters: <object_name>          // string is refDes
```

GetComponentPins (Layout Editor)

Use: Informational.

Command: None.

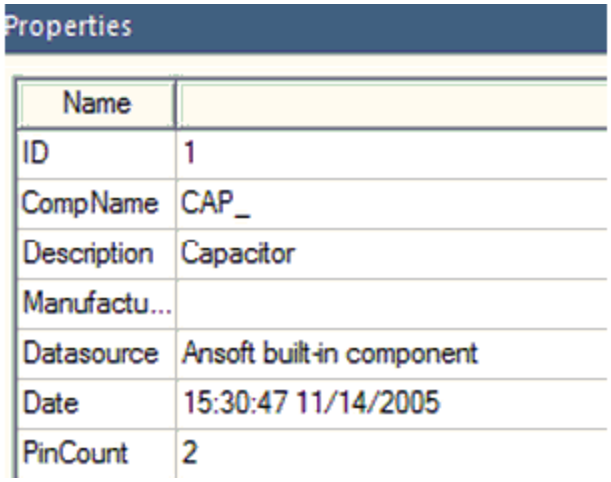
Syntax: GetComponentPins<compID>

Return Value: array of strings, which are the names of all the component's pins

```
Parameters: The LayoutComp's ID
CompInst@<ComponentName>;<CompInstID>
```

Note: For the documented example, the capacitor component name is CAP_ and its ID is 1. If you can select the item of interest in the schematic, the **Properties** window gets updated with the corresponding component name, its ID, and other details as shown in the following figure.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



GetComponentPinInfo (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetComponentPinInfo<compID, pinName>

Return Value: retval = array of strings, as follows:

"X=val"

"Y=val"

"Angle=val"

"Flip=true/false"

"WireID=string"

Parameters: `compID = The LayoutComp's ID`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

pinName = The name of the pin

GetEditorName (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetEditorName()

Return Value: Returns the name of the editor.

GetMaterialList (Layout Editor)

Use: Get the names of all the materials for a layout.

Command: None.

Syntax: GetMaterialList

Return Value: Array of strings.

Parameters: None.

materialNames = oEditor.GetMaterialList

GetProperties (Layout Editor)

Use: Gets a list of all the properties belonging to a specific **PropServer** and **PropTab**. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetProperties(<PropTab>, <PropServer>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: Variant array of strings – the names of the properties belonging to the prop server.

GetPropertyValue (Layout Editor)

Use: Gets the value of a single property. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetPropertyValue(<PropTab>, <PropServer>, <PropName>)

Return Value: String representing the property value.

GetSelections (Layout Editor)

Returns an array of currently selected objects.

UI Access	N/A
Parameters	None.
Return Value	Array containing object IDs

Python Syntax	GetSelections()
Python Example	`oEditor.GetSelections()`

GetSelectionsInHierarchy (Layout Editor)

Returns an array of currently selected objects including the ones inside a subdesign (i.e., an inactive design).

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	Array containing object IDs

Python Syntax	GetSelectionsInHierarchy()
Python Example	`oEditor.GetSelectionsInHierarchy()`

Heal (Layout Editor)

Heals imprecise geometry.

UI Access	From the **Layout** ribbon, click **Healing**.
Parameters	See table below

Name	Type	Description
<HealArray>	Array	Structured array containing these parameters in order: • <NAME:category> • <Type> • <OptionalArgs> • <Tol>
<*NAME:category*>	String	Required first parameter which identifies the category of healing (i.e., "**Snap**", "**Feature**", or "*Repair*", such as <NAME:Snap>.
<*Type*>	String	Dependent on <NAME:category>. Identifies the type of healing, per category: **Snap** — "Point", "Arc", or "Grid" **Feature** — "Voids" or "FloatingBodies" **Repair** — "Colinear" or "SelfIntersecting"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<OptionalArgs>	String	Optional parameters (i.e., "Antipads", "ModifyArcs", and/or "InsertPointsInEdges". Multiple <OptionalArgs> can be entered, depending on <NAME:category> and <Type>.
	<Antipads>	Boolean	If <NAME:Feature> and <Type>="Voids", then this optional argument is available: **True** — voids are converted to antipads **False** — voids are not converted to antipads
	<ModifyArcs>	Boolean	If <NAME:Snap> and <Point>="Voids", then this optional argument is available: **0** — arcs do not snap to close primitive edges **1** — arcs snap to close primitive edges
	<InsertPointsInEdges>	Boolean	If <NAME:Snap> and <Point>="Voids", then this optional argument is available: **0** — new points will not appear when healed primitives snap together **1** — new points will appear at midpoints between existing edges and at the new edges created when healed primitives snap together
	<*Tol*>	Integer	Primitives at or within this tolerance are healed.
Return Value	None		

Python Syntax	oEditor.Heal([<NAME:<category>, <Type>, <OptionalArgs>, <Tol>])

Python Example	`oProject = oDesktop.SetActiveProject("GeometryHealing")`
	`oDesign = oProject.SetActiveDesign("TestDesign")`
	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.Heal(`
	`  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NAME:Snap",
    "Type:=" , "Point",
    "ModifyArcs:=" , 1,
    "InsertPointsInEdges:=" , 1,
    "Tol:=" , "2mm",
    "Tol:=" , "2mm"
  ])
```

PageSetup (Layout Editor)

Use: Specifies page setup for printing.

Command: **File>Page Setup**

Syntax: PageSetup <ArgArray>

Return Value: None.

Parameters: <Margins>: Page margins in implicit units of inches.

<Border>: Integer value indicating to draw border (1) or not to draw (0).

<DesignVars>: Integer value indicating to draw design vars (1) or not to draw (0).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PushExcitations

Allows access to computed excitations for transient and linear frequency solutions. The script command can be accessed from three locations.

<table>
<tr><td>UI Access</td><td>

Layout Editor
Schematic Editor
Select a Nexxim solution in a 3D Layout design, right click, and choose Push Excitations.

</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Reference Designation></td><td>String</td><td>"<refdes>"

This argument is empty when excitations are pushed from a Nexxim solution.</td></tr>
<tr><td><PushExcitations Parameters></td><td>Array</td><td>This parameter changes depending on whether the excitations comes from a transient or linear frequency solution. The keyword <code>"transient:="</code> indicates to AEDT which solution generated the excitations. If <code>"transient:="</code> is present, <code>"CalcThevenin ="</code> and its value are ignored.

For a linear frequency solution, use this array:

<code>["NAME:options",
"CalcThevenin =" , <true|false>,
"Sol:=", "<solution name>"]</code>

If <code>CalcThevenin</code> is true, Thevenin’s equivalent is calculated. Parameters for the linear frequency solution do not include a Freq argument, so all frequencies from the solution are used.

For a transient solution, use:

<code>["NAME:options",
"transient:=",</code></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["start:=", <start time>, "stop:=", <stop time>, "maxHarmonics:=", <max harmonics>, "winType:=", <window>, ["widthPct:=", <width percentage>,] ["kaiser:=", <Kaiser value>,] ["correctCoherentGain:=", true]], "Sol:=", "<solution name>"] winType can have the following values: • Rectangular • Bartlett • Blackman • Hamming • Hanning • Kaiser • Welch • Weber • Lanzcos
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Select (Layout Editor)

Chooses a target list of objects to make active.

UI Access	N/A.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ObjectName*></td><td>Array</td><td>Array consisting of target object names, separated by commas (e.g., rect_0, rect_1, circle_0).</td></tr></table>
Return Value	None.

Python Syntax	Select(<ObjectName>)
Python Example	`oEditor.Select(['rect_0', 'rect_1', 'circle_0'])`

SelectAll (Layout Editor)

Use: Select all elements in the layout editor.

Command: None.

Syntax: SelectAll()

Parameters: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: Dim removedDefs

```
removedDefs = oDefinitionEditor.SelectAll()
```

SetActiveUnits (Layout Editor)

Sets the current active unit of measurement to the target unit.

UI Access	N/A
Parameters	Name | Type | Description <*SelectedUnits*> | Integer | User-specified unit of measurement to set to active (e.g., mm).
Return Value	The previous active unit of measurement.

Python Syntax	SetActiveUnits(<SelectedUnits>)
Python Example	`oEditor.SetActiveUnits('um')` `'mm'`

SetPropertyValue (Layout Editor)

Use: Sets the value of one property. This is not supported for properties of the following types: **ButtonProp, PointProp,** and **VPointProp**. Only the **ChangeProperty** command can be used to modify these properties. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: SetPropertyValue <PropTab>, <PropServer>, <PropName>, <PropValue>

Return Value: None

Parameters: <PropValue>

Type: String

Contains the value to set the property. The formatting is different depending on what type of property is being edited. Use **GetPropertyValue** for the desired property to see the expected format.

Example:
```
oEditor.SetPropertyValue
"BaseElementTab","rect_1",
"LineWidth", "3mm"
```

StitchLines (Layout Editor)

Use: Stitch together connected or crossing lines into polygons and lines.

Command: StitchLines

Syntax: StitchLines Array("NAME:elements", <"line">, …)

Return Value: None

Parameters: <"line">

Type: text

Description: line name.

Example:
```
 oDesign = oProject.SetActiveDesign("HFSS3D1")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oEditor = oDesign.SetActiveEditor("Layout")
oEditor.StitchLines ["NAME:elements", "line_1", "line_2", "line_3"]
```

UnselectAll (Layout Editor)

Use: Unselect all active selections in the layout editor.

Command: Edit > Unselect All

Syntax: UnselectAll()

Parameters: None

Example: `oLayout.UnselectAll()`

ZoomToFit (Layout Editor)

Zoom the current layout to fit the contents of the currently visible page

UI Access	**View** > **Fit All** or the **Fit All** button on the View ribbon.
Parameters	None.
Return Value	None.

Python Syntax	ZoomToFit()
Python Example	`oDesign = oProject.SetActiveDesign("projectName")` `oEditor = oDesign.SetActiveEditor("Layout")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oEditor.ZoomToFit()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

25 - Definition Manager Script Commands

The definition manager controls the use of materials and scripts in a project. It also provides access to the managers for symbols, footprints, padstacks, and components in a project.

```
oProject = oDesktop.SetActiveProject("Project1")

oDefinitionManager = oProject.GetDefinitionManager()
```

The topics for this section include:

Add (ComponentManager)

AddDataset

AddMaterial

Add (Footprint Manager)

Add (PadstackManager)

Add (SymbolManager)

AreMaterialPropertiesEqual

AreSurfaceMaterialPropertiesEqual

ChangeProperty

CloneMaterial

ComputeCoreLossCoefficients

DeleteDataset

DoesMaterialExist

Edit (ComponentManager)

EditDataset

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditMaterial

Edit (PadstackManager)

ExportDataset

ExportFootprintManager

ExportMaterial

Export (PadstackManager)

ExportScript

Export (SymbolManager)

GetProjectMaterialNames

GetPropertyValue

Remove (ComponentManager)

Remove (FootprintManager)

RemoveMaterial

Remove (PadstackManager)

RemoveScript

Remove (SymbolManager)

SetPropertyValue

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Related Topics:

Component Manager Script Commands

Symbol Manager Script Commands

Material Manager Script Commands

Model Manager Script Commands

Network Data Explorer Manager Script Commands

Script and Library Scripts

Add [component manager]

Add creates a new component in the component library.

UI Access	**Tools** > **Edit Libraries** > **Component** > **Add**		
Parameters	Name	Type	Description
	<NAME>	string	Component name.
	<Info>		Lists the component information described below.
	<Type>	int	Component type.
	<NumTerminals>	int	Number of terminals.
	<DataSource>	string	Data source specified.
	<ModifiedOn>	int	Date of last change to the component.
	<Manufacturer>	string	Component manufacturer.
	<Symbol>	string	Symbol used to represent the component.
	<Component>	string	Component
	<ModelNames>	string	Component model name
	<Footprint>	string	Component footprint.
	<Description>	string	Component description.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<InfoTopic>	string	Component information topic file.
	<InfoHelpFile>	string	Path to the help file.
	<IconFile>	string	Path to the component icon.
	<Library>	string	Component library.
	<OriginalLocation>	string	Original location of the component.
	<Project>	string	Project of the component.
	<IEEE>	string	Component IEEE information.
	<Author>	string	Most recent editor of the component.
	<OriginalAuthor>	string	Original creator of the component.
	<CreationDate>	string	Date of component when first created.
	<ExampleFile>	string	Path to component example file.
	<HiddenComponent>	int	Number of hidden components.
	<GroupID>	string	Identification tag of the component group.
	<Refbase>	string	Component reference designator.
	<NumParts>	int	Number of parts in the component.
	<ModSinceLib>	bool	True or False
	<CompExtID>	int	External component identification number.
Return Value	None		

Python Syntax	oComponentManager.Add([ComponentInfo])

Python Example	`oComponentManager.Add(`
	`  ["NAME:Component",`
	`  "Info:=", ["Type:=", 0,`
	`  "NumTerminals:=", 0,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "DataSource:=" , "",
        "ModifiedOn:=" , 1747683313,
        "Manufacturer:=", "","Symbol:=", "Component", "ModelNames:=", "",
        "Footprint:=",
        "Description:=", "",
        "InfoTopic:=", "",
        "InfoHelpFile:=", "",
        "IconFile:=", "",
        "Library:=" , "",
        "OriginalLocation:=" , "Project",
        "IEEE:=" , "", "
        Author:=" , "",
        "OriginalAuthor:=" , "",
         "CreationDate:=" , 1747683307,
         "ExampleFile:=" , "",
        "HiddenComponent:=" , 0,
        "CircuitEnv:=" , 0,
        "GroupID:=" , 0
    ],
    "CircuitEnv:=", 0,
    "Refbase:=" , "U",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NumParts:=" , 1, "ModSinceLib:=" , True, "CompExtID:=" , 1])

AddDataset

Adds a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Add**
Parameters	Name / Type / Description <*DatasetDataArray*> / Array / ["NAME:<DatasetName>", ["NAME:Coordinates", <CoordinateArray>, <CoordinateArray>, ...] <*DatasetName*> / String / Name of the dataset. <*CoordinateArray*> / Array / ["NAME:Coordinate", "X:=", <double>, "Y:=",<double>]
Return Value	None.

Python Syntax	AddDataset <*DatasetDataArray*>
Python Example	oProject.AddDataset(["NAME:$ds1", ["NAME:Coordinates", ["NAME:Coordinate",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "X:=", 2,
        "Y:=", 4
      ],
      ["NAME:Coordinate",
        "X:=", 6,
        "Y:=", 8
      ]
    ]
]
)
oDesign.AddDataset(
[
"NAME:$ds1",
  ["NAME:Coordinates",
    ["NAME:Coordinate",
      "X:=", 2,
      "Y:=", 4
    ],
    ["NAME:Coordinate",
      "X:=", 6,
      "Y:=", 8
    ]
  ]
]
)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddDefinitionFromBlock

Adds a material definition from block text (same definition format as would be contained in the material library file) by library type (using definition folder name). This scripting command directly supports the .AMAT (or .ASURF) definition formats.

UI Access	N/A
Parameters	

Name	Type	Description
<*defBlock*>	String	Text of the new material definition in block form.
<*defFolderName*>	String	Library type (by definition folder name)
<*newTimeStamp*>	String	New timestamp (time_t as integer number of seconds since 1/1/1970 12:00am, as string), default is current time
<*replaceExisting*>	Boolean	True to replace existing, False to choose a new unique name if an existing definition is found

Return Value	A property scripting object for the definition.

Python Syntax	AddDefinitionFromBlock(<*defBlock*>,<*defFolderName*>, <*newTimeStamp*>, <*replaceExisting*>)
Python Example	

```
oProject = oDesktop.NewProject()
oProject.InsertDesign("HFSS", "HFSSDesign1", "DrivenModal", "")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateBox(
  ["NAME:BoxParameters",
    "XPosition:=" , "-0.4mm",
    "YPosition:=" , "-1mm",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ZPosition:=" , "0mm",
    "XSize:=" , "1.4mm",
    "YSize:=" , "1.6mm",
    "ZSize:=" , "0.6mm"
  ],
  ["NAME:Attributes",
    "Name:=" , "Box1",
    "Flags:=" , "",
    "Color:=" , "(143 175 143)",
    "Transparency:=" , 0,
    "PartCoordinateSystem:=", "Global",
    "UDMId:=" , "",
    "MaterialValue:=" , "\"vacuum\"",
    "SurfaceMaterialValue:=", "\"\"",
    "SolveInside:=" , True,
    "ShellElement:=" , False,
    "ShellElementThickness:=", "0mm",
    "IsMaterialEditable:=" , True,
    "UseMaterialAppearance:=", False,
    "IsLightweight:=" , False
  ])
oDefinitionManager = oProject.GetDefinitionManager()
defBlock = "$begin 'vacuum2' $begin 'AttachedData' $begin 'MatAppearanceData'
property_data='appearance_data' Red=230 Green=230 Blue=230 Transparency=0.95
$end 'MatAppearanceData' $end 'AttachedData' simple('permittivity', 1)
ModTime=1499970477 $end 'vacuum2'"
added = oDefinitionManager.AddDefinitionFromBlock(defBlock, "Materials",
"10101010", True)
addedName = ''
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`  if isinstance(added, basestring):`
	`addedName = added`
	`  elif isinstance(added, list):`
	`addedName = added[0]`
	`else:`
	`  addedName = added.GetName().replace("Materials:", "")`
	`AddInfoMessage(os.path.basename(__file__) + " result: " + addedName)`
	`materialNameInQuotes = "\"" + addedName + "\""`
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`    ["NAME:Geometry3DAttributeTab",`
	`      ["NAME:PropServers",`
	`        "Box1"`
	`      ],`
	`      ["NAME:ChangedProps",`
	`        ["NAME:Material",`
	`         "Value:=", materialNameInQuotes`
	`        ]`
	`      ]`
	`    ]`
	`  ])`

AddMaterial

Adds a local material.

UI Access	**Add Material** in the material editor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*MaterialParams*>	Array	["NAME: <name of the material to be added>", <MatProperty>, <MatProperty>, ...]
<*MatProperty*>	Array	For simple material: "<PropertyName>:=", <value> For anisotropic material: ["NAME:<PropertyName>", "property_type:=", "AnisoProperty", "unit:=", <Unit>", "component1:=", <value>, "component2:=", <value>, "component3:=", <value>))]
<*PropertyName*>	String	Should be one of the following (depending on the material, design, and solution types):

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Electromagnetic (Maxwell-exclusive material properties omitted, see Maxwell Scripting help): "permittivity", "permeability", "conductivity", "dielectric_loss_tangent", "magnetic_loss_tangent", "electric_coercivity", "magnetic_coercivity", "saturation_mag", "lande_g_factor", "delta_H", "delta_h_freq", "mass_density" Thermal (including solids, Icepak fluid flow, and IcepakFEA rotating fluid modeling): "thermal_conductivity", "mass_density", "specific_heat", "thermal_expansion_coefficient", "thermal_material_type", "viscosity", "diffusivity", "molecular_mass", "clarity_type" Structural: "mass_density", "youngs_modulus", "poissons_ratio", "thermal_expansion_coefficient"
	<*Unit*>	String	Possible values (Maxwell-exclusive properties omitted, see Maxwell Scripting Help; other missing entries are unitless): conductivity: "siemens/m" saturation_mag: "uTesla", "mTesla", "tesla", "kTesla", "uGauss", "mGauss", "gauss", "kGauss" delta_H: "A_per_meter", "kA_per_meter", "Oe", "kOe" delta_h_frequency: "Hz", "kHz", "MHz", "GHz", "THz", "rps", "per_sec"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			mass_density: "kg/m^3" thermal_conductivity: "W/m-C" specific_heat: "J/kg-C" youngs_modulus: "N/m^2" thermal_expansion_coefficient: "1/C"
Return Value	None		

Python Syntax	AddMaterial (["NAME:<MaterialName>", <MatProperty>, <MatProperty>, ...])

Python Example	`oDefinitionManager.AddMaterial(`
	`["permittivity:=", "2.2", "0.002"])`
	
	`oDefinitionManager.AddMaterial(`
	`["NAME:Material2",`
	`  "dielectric_loss_tangent:=", "44",`
	`  ["NAME:saturation_mag",`
	`  "property_type:=", "AnisoProperty",`
	`  "unit:=", "Gauss",`
	`  "component1:=", "11",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "component2:=", "22",
    "component3:=", "33"],
  "delta_H:=", "44Oe"])
```

Add [padstack manager]

Adds a padstack.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Add Padstack**		
Parameters	Name	Type	Description
	<NAME>	String	Simple name of padstack to create; in the form of "Name:PadstackName"
	<ModifiedOn>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
	<Library>	String	Padstack library.
	<LibLocation>	String	Location of the padstack.
	<psdArray>	Array	Includes "Name:psd" and the subparameters listed below.
	<nam>	String	<PadstackName>
	<lib>	String	Name of the library
	<mat>	String	Hole plating material
	<plt>	String	Percent of hole's radius filled by plating
	<pdsArray>	Array	Includes "Name:pds" and a list of *<LayerGeometryArray>* that defines the padstack geometry. See below for more details this array.
	<hle>	Array	Defines <PadInfo>; see below for more details on this array.
	<hRg>	String	Defines the hole range. One of the following choices:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Thr" – through all layout layers "Beg" – from upper pad layer to lowest layout layer "End" – from upper layout layer to lowest pad layer "UTL" – from upper pad layer to lowest pad layer
	<sbsh>	String	Defines the solder ball shape. One of the following choices: "None" – no solderball "Cyl" – cylinder solderball "Sph" – spheroid solderball
	<sbpl>	String	Defines the solder ball placement. One of the following choices: "abv" – above padstack "blw" – below padstack
	<sbr>	String	Solderball diameter; real number with units
	<sb2>	String	Solderball mid diameter, real number with units
	<sbn>	String	Name of solderball material
	<PadPortLayerArray>	Array	List of integers where each integer is a layer id
Return Value	Simple name of the added padstack. If the name requested conflicts with the name of an existing padstack, the requested name is altered to be unique, and the name returned reflects any change made.		

<LayerGeometryArray>:

```
["Name:lgm",
"lay:=", <string>, // definition layer name
"id:=", <int>, // definition layer id
"pad:=", <PadInfo>, // pad
"ant:=", <PadInfo>, // antipad
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"thm:=", <PadInfo>, // themal pad
"X:=", <string>, // pad x connection, real with units
"Y:=", <string>, // pad y connection, real with units
"dir:=", <DirectionString>] // pad connection direction
```

<PadInfo>:

```
["shp:=", <PadShape>,
"Szs:=", <DimensionArray>,
"X:=", <string>, // x offset, real with units
"Y:=", <string>, // y offset, real with units
"R:=", <string>] // rotation, real with units
```

<PadShape>:

One of these choices:

```
"No" // no pad
"Cir" // Circle
"Sq" // Square
"Rct" // Rectangle
"Ov" // Oval
"Blt" // Bullet
"Ply" // Polygons
"R45" // Round 45 thermal
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"R90" // Round 90 thermal
"S45" // Square 45 thermal
"S90" // Square 90 thermal
```

<DimensionArray>:

An array of strings; each string is a real with units for one of the dimensions of the shape

<DirectionString>:

One of these choices:

```
"No" // no direction
"Any" // any direction
"0" // 0 degrees
"45" // 45 degrees
"90" // 90 degrees
"135" // 135 degrees
"180" // 180 degrees
"225" // 225 degrees
"270" // 270 degrees
"315" // 315 degrees
```

Python Syntax	oPadstackManager.Add(<*PadstackName*>, <*ModifiedOnInfo*>, <*psdArray*>,<PadPortLayerArray>)

Python Example	`oPadstackManager.Add(`
	`  ["NAME:Circle - through3",`
	`  "ModTime:=", 1235765635,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Library:=", "",
"LibLocation:=", "Project",
["NAME:psd", "nam:=", "Circle - through3", "lib:=", "", "mat:=", "",
"plt:=", "0",
  ["NAME:pds",
    ["NAME:lgm", "lay:=", "Top Signal", "id:=", 0,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2.5mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", ["shp:=", "Cir", "Szs:=", ["3.5mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0"],
      "thm:=", ["shp:=", "No", "Szs:=", [], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0"],
       "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
    ["NAME:lgm", "lay:=", "SignalA", "id:=", 1,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", ["shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0"],
      "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0"],
      "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
    ["NAME:lgm", "lay:=", "SignalB", "id:=", 2,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", [shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
      ["NAME:lgm", "lay:=", "Ground", "id:=", 3,
        "pad:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0deg"],
        "ant:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0"],
        "thm:=", ["shp:=", "R90", "Szs:=", ["3mm", "0.75mm", "1mm"], "X:=",
        "0mm", "Y:=", "0mm", "R:=", "0"],
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
      ["NAME:lgm", "lay:=", "Bottom signal", "id:=", 5,
        "pad:=", ["shp:=", "Cir", "Szs:=", ["1mm"], "X:=", "0mm", "Y:=",
        "0mm", "R:=", "0deg"],
        "ant:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
        "0mm", "R:=", "0deg"],
        "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0"],
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"]
    ]
    "hle:=", ["shp:=", "Cir", "Szs:=", ["1.5mm"], "X:=", "0mm", "Y:=", "0mm",
    "R:=", "0deg"],
    "hRg:=", "End",
    "sbsh:=", "Sph",
    "sbpl:=", "abv",
    "sbr:=", "750um",
    "sb2:=", "1200um", "1200um",
    "sbn:=", "solder"],
  "ppl:=", [0, 1, 2, 3, 5]
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Add [symbol manager]

Use: Add a symbol

Command: Tools > Edit Configured Libraries > Symbols > Add Symbol

Syntax: Add Array("NAME:<SymbolName>",

"ModTime:=", <ModifiedTimeInfo>,

"Library:=", "", // Library name

"LibLocation:=", "Project", // Project Location

<PinDefInfo>,

<PinDefInfo>,... // optional, to define pins

<GraphicsDataInfo>, // optional, to define graphics

<PropDisplayMapInfo>)) // optional, to define property displays

Return Value: <string>

// composite name of the symbol.

// If the name requested conflicts with the name of an existing

// symbol, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <SymbolName>:

<string> // simple name of the symbol being added

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<PinDefInfo>:

Array("NAME:PinDef",
"Pin:=", Array (<string>, // pin name
<real>, // x location
<real>, // y location
<real>, // angle in radians
<PinType>,
<real>, // line width
<real>, // line length
<bool>, // mirrored
<int>, // color
<bool>, // true if visible, false if not
<string>, // hidden net name
<OptionalPinInfo>, // optional info
<PropDisplayMapInfo>)) // optional

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<PinType>:

<string> // "N" : normal pin

// "I" : input pin

// "O" : output pin

<OptionalPinInfo>:

// Specify both or neither

<bool>, // true if name is to be shown

<bool>, // true if number is to be shown

<PropDisplayMapInfo>:

Array("NAME:PropDisplayMap",

<PropDisplayInfo>,

<PropDisplayInfo>,...)

<PropDisplayInfo>:

<NameString>, Array(<DisplayTypeInfo>,

<DisplayLocationInfo>,

<int>, // optional, level number

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<TextInfo>)

<NameString>:

<string> // PropertyName:=, where PropertyName is the name of

// the property to be displayed

<DisplayTypeInfo>:

<int> // 0 : No display

// 1 : Display name only

// 2 : Display value only

// 3 : Display both name and value

// 4: Display evaluated value only

// 5: Display both name and evaluated value

<DisplayLocationInfo>:

<int> // 0 : Left

// 1 : Top

// 2 : Right

// 3 : Bottom

// 4 : Center

// 5 : Custom placement

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<GraphicsDataInfo>:

Array("NAME:Graphics",

// one or more of the following

<RectInfo>,

<CircleInfo>,

<ArcInfo>,

<LineInfo>,

<PolygonInfo>,

<TextInfo>,

<ImageInfo>)

<RectInfo>:

"Rect:=", Array(<real>, // line width

<int>, // fill pattern

<int>, // color

<real>, // angle, in radians

<real>, // x position of center

<real>, // y position of center

<real>, // width

< real>) // height

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<CircleInfo>:

"Circle:=", Array(<real>, // line width

<int>, // fill pattern

<int>, // color

<real>, // x position of center

<real>, // y position of center

< real>) // radius

<ArcInfo>:

"Arc:=", Array(<real>, // line width

<int>, // line pattern

<int>, // color

<real>, // x position of center

<real>, // y position of center

< real>, // radius

<real>, // start angle, in radians

<end>) // end angle, in radians

<LineInfo>:

"Line:=", Array(<real>, // line width

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<int>, // line pattern

<int>, // color

<PointInfo>, // must specify at least 2 points

<PointInfo>...)

<PointInfo>:

<real>, // x position

<real> // y position

<PolygonInfo>:

"Polygon:=", Array(<real>, // line width

<int>, // fill pattern

<int>, // color

<PointInfo>, // must specify at least 3 points

<PointInfo>...)

<TextInfo>:

"Text:=", Array(<real>, // x position

<real>, // y position

<real>, // angle, in radians

<Justification>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<bool>, // is plotter font
<string>, // font name
<int>, // color
<string>) // text string

<Justification>:
<int> // 0 : left top
// 1 : left base
// 2 : left bottom
// 3 : center top
// 4 : center base
// 5 : center bottom
// 6 : right top
// 7 : right base
// 8 : right bottom

<ImageInfo>:
"Image:=", Array(<RectInfo>,
<ImageData>,
<bool>) // is mirrored

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<ImageData>:

<string>, // file path

<int>, // 0 : use the file path and link to it

// 1 : ignore file path and use next parameter

<string> // text data, only present if preceding int is 1

AreMaterialPropertiesEqual

Checks whether named material compared to added material data have equivalent major properties.

<table>
<tr><td>UI Access</td><td>None.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>material_data</td><td><string></td><td>This will be the same format as the AddMaterial() input parameter, material_data. For example, material_data can be something like this.["NAME:Mold_Material", "CoordinateSystemType:=", "Cartesian", "BulkOrSurfaceType:=", 1, ["NAME:PhysicsTypes", "set:=", ["Thermal"]], "thermal_conductivity:=", "0.8", "mass_density:=", "1980", "specific_heat:=", "960"].</td></tr>
<tr><td><<i>MaterialName</i>></td><td><String></td><td>Name of the material to compare with material database</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Boolean:
True – named materials have equivalent major properties.
False – named materials do not have equivalent major properties.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AreMaterialPropetiesEqual(<material_data>, '<*MaterialName*>')
Python Example	`oDefinitionManager.AreMaterialPropertiesEqual(material_data, 'Mold_Material')`

AreSurfaceMaterialPropertiesEqual

Checks whether named surface material compared to added surface material data have equivalent major properties.

UI Access	None.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>material_data</td><td><string></td><td>This will be the same format as the AddMaterial() input parameter, material_data. For example, material_data can be something like this.["NAME:Mold_Material", "CoordinateSystemType:=", "Cartesian", "BulkOrSurfaceType:=", 1, ["NAME:PhysicsTypes", "set:=", ["Thermal"]], "thermal_conductivity:=", "0.8", "mass_density:=", "1980", "specific_heat:=", "960"]</td></tr><tr><td><*MaterialName*></td><td><String></td><td>Name of the surface material to compare with material database</td></tr></table>
Return Value	Boolean: • **True** – named surface materials have equivalent major properties. • **False** – named surface materials do not have equivalent major properties.

Python Syntax	AreSurfaceMaterialPropetiesEqual(<material_data>, '<*MaterialName*>')

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDefinitionManager.AreSurfaceMaterialPropertiesEqual(material_data, 'Mold_Material')`

ChangeProperty (Symbol Editor)

Use: Changes to properties are scripted using the **ChangeProperty** command. This command can be executed by the **oEditor** to change editor properties, by the **oDesign** to change design level properties, and by the **oProject** to change project level properties. The command can be used to create, edit, and/or remove properties. In Circuit, only Variable and Separator properties can be deleted.

Command: None

Syntax: ChangeProperty Array("Name:AllTabs", <PropTabArray>, <PropTabArray>, ...)

Return Value: None

Parameters: <PropTabArray>

Array("Name:<PropTab>",

<PropServersArray>,

<NewPropsArray>,

<ChangedPropsArray>,

<DeletedPropsArray>)

<PropServersArray>

Array("Name:PropServers", <PropServer>,

<PropServer>, ...)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<NewPropsArray>
Array("Name:NewProps", <PropDataArray>,
<PropDataArray>,...)

<ChangedPropsArray>
Array("Name:ChangedProps",<PropDataArray>,
<PropDataArray>, ...)

<DeletedPropsArray>
Array("Name:DeletedProps", <PropName>,
<PropName>, ...)
OR (for PropDisplay deletions only)
Array("Name:DeletedProps",<PropDataArray>,
<PropDataArray>, ...)

<PropDataArray>
Array("NAME:<PropName>",
"PropType:=", <PropType>,
"NewName:=", <string>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Description:=", <string>,

"Callback:=", <string>,

"NewRowPosition:=", <int>,

"ReadOnly:=", <bool>,

"Hidden:=", <bool>,

<PropTypeSpecificArgs>)OR (for PropDisplays only)

Array("Name:<PropName>",<PropDisplayData>)

<PropDisplayData>

for adding, changing, deleting PropDisplays

<PropDisplayAttributes>

for changing PropDisplays only

<PropDisplayNewAttributes>

<PropDisplayAttributes>

Layer & Location only used for PropDisplays in layout

For adding PropDisplays, this will add a single PropDisplay with attributes as shown; if an attribute is missing, a default value will be assigned. Adding PropDisplay to schematic with attributes that are identical to one already existing there will fail without an error message.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

For deleting PropDisplays, these attributes are used to identify an existing PropDisplay to delete. If there doesn't exist a PropDisplay that matches the given attributes, then nothing will be deleted. If multiple PropDisplays match the given attributes, then all of them will be deleted. If an attribute is missing, then all PropDisplays match that missing attribute. For example, if Layer is missing, then PropDisplays on all layers that match the remaining given attributes will be deleted.

For changing PropDisplays, these attributes are used to identify an existing PropDisplay to change. If no PropDisplay matching the attributes is found, no changes will be made. If multiple PropDisplays match the attributes, all of them will be changed. If an attribute is missing, it matches all PropDisplays. For example, to change the format of PropDisplays that are on the bottom, but have any layer, style or format to show the name only, this command should have Location set to "Bottom" and all other attributes omitted.

"Format:=", <PropDisplayType>,

"Location:=", <PropDisplayLocation>,

"Layer:=", <string>,

"Style:=", <string>

<PropDisplayNewAttributes>

NewLayer & NewLocation only used for PropDisplays in layout

For changing PropDisplays, these attributes are used to identify which attributes to change and what the new value is. If the attribute should not be changed, the corresponding entry should be omitted.

"NewName:=", <string>,

"NewFormat:=", <PropDisplayType>,

"NewLocation:=", <PropDisplayLocation>,

"NewLayer:=", <string>,

"NewStyle:=", <string>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<PropDisplayType>

Type: string

Identifies the format of PropDisplay.

"Name"

"Value"

"NameAndValue"

"EvaluatedValue"

"NameAndEvaluatedValue"

<PropDisplayLocation>

Type: string

Identifies where PropDisplay is located with respect to object

"Left"

"Top"

"Right"

"Bottom"

"Custom"

<PropType>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: string

Identifies the type of property when a new property is added. In Circuit, only separator properties and variable properties can be added.

"SeparatorProp"

"VariableProp"

"TextProp"

"NumberProp"

"ValueProp"

"CheckboxProp"

"MenuProp"

"PointProp"

"VPointProp"

"ButtonProp"

NewName

Specify the new name of a property if the property's name is being edited. In Circuit, the name can only be changed for separators and variables.

Description

Specify a description of the property. In Circuit, the description can only be changed for separators and variables.

Callback

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Specify the name of the script callback to be run when the property value is changed.

NewRowPosition

Used to reorder rows in the **Property** dialog box. In Circuit, this only applies to the **Project>Project Variables** panel and the **Designer>DesignProperties** panel. Specify the new zero-based row index of the variable or separator.

ReadOnly

Used to mark a property as "read only" so it can not be modified. In Circuit, this flag can only be set for variables and separators.

Hidden

Used to hide a property so it can not be viewed outside of the **Property** dialog box. In Circuit, this flag can only be set for variables and separators.

<PropTypeSpecificArgs>

SeparatorProp: no arguments

TextProp: "Value:=", <string>

NumberProp: "Value:=", <double>

ValueProp: "Value:=", <value>

CheckboxProp: "Value:=", <bool>

MenuProp: "Value:=", <string>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PointProp "X:=", <double>, "Y:=", <double>

VPointProp: "X:=", <value>, "Y:=", <value>

Material Button: "Material:=", <string>

Color Button: "R:=",<int>,"G:=",<int>,"B:=",<int>

Transparency Button: "Value:=", <double>

<PropTypeSpecificArgs> for MenuProps

Syntax for NewProps array: "AllChoices:=",

<"choice1,choice2,..."> or <Array("choice1" "choice2", ...)>,

"Value:=", <string>

Syntax for ChangedProps array: "Value:=", <string>

<PropTypeSpecificArgs> for VariableProps

Syntax:

"Value:=", <value>, <OptimizationFlagsArray>,

<TuningFlagsArray>, <SensitivityFlagsArray>,

<StatisticsFlagsArray>

Parameters:

<OptimizationFlagsArray>

Array("NAME:Optimization",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Included:=", <bool>,

"Min:=", <value>,

"Max:=", <value>)

<TuningFlagsArray>

Array("NAME:Tuning",

"Included:=", <bool>,

"Step:=", <value>,

"Min:=", <value>,

"Max:=", <value>)

<SensitivityFlagsArray>

Array("NAME:Sensitivity",

"Included:=", <bool>,

"Min:=", <value>,

"Max:=", <value>,

"IDisp:=", <value>)

<StatisticsFlagsArray>

Array("NAME:Statistical",

"Included:=", <bool>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Dist:=", <Distribution>,

"StdD:=", <value>,

"Min:=", <value>,

"Max:=", <value>,

"Tol:=", <string>)

<Distribution>

Type: string

Value should be **"Gaussian"** or **"Uniform"**

StdD

Standard deviation.

Min

Low cut-off for the distribution.

Max

High cut-off for the distribution.

Tol

Tolerance for uniform distributions. Format is **"<int>%"**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: **"20%"**.

CloneMaterial

Clones a local material.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*matName*></td><td>String</td><td>Name of existing material.</td></tr><tr><td><*newName*></td><td>String</td><td>Name for newly cloned material.</td></tr></table>
Return Value	Boolean: • 1 - Material is cloned. • 0 - Existing material not found or a conflict with the new material name.

Python Syntax	CloneMaterial (<*matName*>, <*newName*>)
Python Example	`oDefinitionManager.CloneMaterial("copper1", "copper3")`

DeleteDataset

Deletes a specified dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Remove**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*DatasetName*>	String	Name of the dataset found in the project.
Return Value	None.		

Python Syntax	DeleteDataset (<*DatasetName*>)
Python Example	`oProject.DeleteDataset('$ds1')` `oDesign.DeleteDataset('$ds1')`

DoesMaterialExist

Checks for the presence of a material in the library by name

UI Access	None.		
Parameters	Name	Type	Description
	<*MaterialName*>	<String>	Name of the material to search for in the material database
Return Value	Boolean: • **True** – specified material exists. • **False** – specified material does not exist.		

Python Syntax	DoesMaterialExist(<*MaterialName*>)
Python Example	`oDefinitionManager.DoesMaterialExist("modified_epoxy")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit [component manager]

Modifies an existing component.

<table>
<tr><td>UI Access</td><td colspan="3">Tools > Edit Configured Libraries > Components > Edit Component</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td></td></tr>
<tr><td><ComponentName></td><td>String</td><td>Name of component to edit</td></tr>
<tr><td><NewComponentName></td><td>String</td><td>Name of the new component created from the edits</td></tr>
<tr><td><Info></td><td>Array</td><td>Provides the component info. It includes the following:

"Type:=", <TypeInfo>,# see more information below
"NumTerminals:=", <int>,
"DataSource:=", <string>,
"ModifiedOn:=", <ModifiedOnInfo>,
"Manufacturer:=", "<string>,
"Symbol:=", <string>,
"Footprint:=", <string>,
"Description:=", <string>,
"InfoTopic:=", <string>,
"InfoHelpFile:=", <string>,
"IconFile:=", <string>,
"LibraryName:=", <string>,
"OriginalLocation:=", <string>, // Project Location
"Author:=", <string>,
"OriginalAuthor:=", <string>,
"CreationDate:= ", <int>)</td></tr>
<tr><td><RefBase></td><td>String</td><td>Reference designator</td></tr>
<tr><td><NumParts></td><td>Integer</td><td>Parts per component</td></tr>
<tr><td><OriginalComponent></td><td>String</td><td>Name of the original component</td></tr>
<tr><td><Terminal></td><td>Array</td><td></td></tr>
<tr><td><Parameters></td><td>Array</td><td>Optional; defines the parameters of the new component. It can include</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			any or all of the following:
			`"VariableProp:=", <VariableInfo>,`
			`"CheckboxProp:=", <CheckBoxInfo>,`
			`"ButtonProp:=", <ButtonInfo>,`
			`"TextProp:=", <TextInfo>,`
			`"NumberProp:=", <NumberInfo>,`
			`"SeparatorProp:=", <SeparatorInfo>,`
			`"ValueProp:=", <ValueInfo>,`
			`"MenuProp:=", <MenuInfo>`

	<Properties>	Array	Optional; defines the properties of the new component. It can include any or all of the following:
			`"CheckboxProp:=", <CheckBoxInfo>,`
			`"TextProp:=", <TextInfo>,`
			`"NumberProp:=", <NumberInfo>,`
			`"SeparatorProp:=",<SeparatorInfo>,`
			`"ValueProp:=", <ValueInfo>,`
			`"MenuProp:=", <MenuInfo>),`
			`"VPointProp:=", <VPointInfo>,`
			`"PointProp:=", <PointInfo>`
			The array structure for each property is defined below.

	<Quantities>	Array	`("Quantities","QuantityProp:=", <Quant-ityPropInfo>...)` The <QuantityPropInfo> array is defined below.
	<CosimDefInfo>	Array	`("NAME:CosimDefinitions", <CosimDe-fInfo>,<CosimDefInfo>...)` The<CosimDefInfo> array is defined below.
Return Value	The composite name of the component. If the name requested conflicts with the name of an existing component, the requested name is altered to be unique. The name returned reflects any change made to be unique.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<TypeInfo>:

An integer that is the or-ing of bits for each product listed below. The default setting is 0xffffffff (4294967295), which indicates valid for all products. In the component editing dialog box, checking different boxes in the "Specify products for which this component is valid" grid control sets specific flags that correspond to the following hex/decimal settings:

```
Nexxim – 100 binary, 4 decimal, 0x4

SIwaveDeNovo – 1000 binary, 8 decimal, 0x8

Simplorer – 10000 binary, 16 decimal, 0x10

MaxwellCircuit – 100000 binary, 32 decimal, 0x20
```

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<TerminalInfo>:

```
[<string>, // symbol pin
<string> // footprint pin
<string >, // gate name
<bool>, // shared
<int>, // equivalence number
<int>, // what to do if unconnected: flag as error:0, ignore:1
<string>, // description
<Nature>]

<Nature>:
<string> // content varies as follows

Nexxim/Circuit:
"Electrical" // the only choice
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Simplorer:
# choices include "Electrical", "Magnetic", "Fluidic", "Translational",
"Translational_V", "Rotational", "Rotational_V",
""Radiant", "Thermal", or <VHDLPackageName>

<VHDLPackageName>:
<string> # in the form <Library>.<Package>
<Library>:
<string> # name of the VHDL library
<Package>:
<string> # name of the VHDL package
```

<VariableInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: number, variable, or expression

<FlagLetters>:
<string> # "D" - has description parameter,
# "RD" - readonly & has description parameter,
# or "RHD" - readonly, hidden, & has description parameter
```

<CheckBoxInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<bool>] # value: true or false
```

<ButtonInfo>:

```
[<string>, # name
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<FlagLetters>,
<string>, # description
<string>, # button title
<string>, # extra text
<ClientID>,
 "ButtonPropClientData:= ", <ClientDataArray>]

<ClientID>:
<int> # specifies Button Prop Client:
# 0 - unknown, "ButtonPropClientData" array will be empty
# 1 - Netlist Prop Client
# 2 - not used
# 3 - File Name Prop Client

<ClientDataArray>:
varies with <ClientID>

<ClientID> is 0 or 1: empty array[]
<ClientID> is 3:
["InternalFormatText:=", "<prefix><RelativePath>"]
<prefix>:
<string> # "<Project>", "<PersonalLib>", "<UserLib>", or "<SysLib>"
<RelativePath>:
<string> # relative path to file from <prefix>
```

<TextInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a text string
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<NumberInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<real>, # value: a number
<string>] # units
```

<SeparatorInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a text string
```

<ValueInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a number, variable or expression
```

<MenuPropInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
<string>, # menu choices - separated by commas
<int>] # 0 based index of current menu choice
```

<VPointInfo>:

```
[<string>, # name
<FlagLetters>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>, # x value: number with length units
<string>] # y value: number with length units
```

<PointInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<real>, # x value
<real>] # y value
```

<QuantityPropInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
<string>, # value
<TypeString>, # string, options are "Across", "Through", or "Free"
<TypeStringDependentInfo>]

<TypeStringDependentInfo>:
"Free" :
<string>, # direction: "In", "Out", "InOut", or "DontCare"
# Following <string> is not present if direction is "DontCare"
<string> # when to calculate: "BeforeAnalogSolver",
// "BeforeStateGraph", "AfterStateGraph", or "DontCareWhen"
"Across" or "Through"
<int>, # terminal 1
<int> # terminal 2
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<CosimDefInfo>:

```
["NAME:CosimDefinition",
"CosimulatorType:=", <int>,
"CosimDefName:=", <string> # "HFSS3D", "Circuit", "Custom", or "Netlist"
"IsDefinition:=", <bool>,
final array members]
```

Final array member(s) vary with CosimDefName:

- final array members for HFSS 3D Layout:

  ```
  "CosimStackup:=", <string>,
  "CosimDmbedRatio:=", <int>
  ```

- final array members for Circuit:

  ```
  "ExportAsNport:=", <int>,
  "UsePjt:=", <int>
  ```

- final array member for Custom:

  ```
  "DefinitionCompName:=", <string>
  ```

- final array member for Netlist:

  ```
  "NetlistString:=", <string>
  ```

Python Syntax	Edit (<*ComponentName*>, [<*NewComponentName*>", <*Info*>, <*RefBase*>,<*NumParts*>, <*Terminals*>, <*Parameters*>,<*Properties*>, <*Quantities*>,<*CosimDefInfo*>])

Python Example	`name = oComponentManager.Edit ("Simplorer Circuit Elements\BJTs:Level01_NPN",`
	`["NAME:Level01_NPN",`
	`"Info:=", ["Type:=", 4294901764,`
	`"NumTerminals:=", 3,`
	`"DataSource:=", "Ansoft built-in component",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "ModifiedOn:=", 1152722112,
      "Manufacturer:=", "",
      "Symbol:=", "nexx_bjt_npn",
      "Footprint:=", "",
      "Description:=", "BJT, GP, NPN",
      "InfoTopic:=", "NXBJT1.htm",
      "InfoHelpFile:=", "nexximcomponents.chm",
      "IconFile:=", "bjtsn.bmp",
      "Library:=", "Nexxim Circuit Elements\BJTs",
      "OriginalLocation:=", "SysLibrary ",
      "Author:=", "",
      "OriginalAuthor:=", "",
      "CreationDate:=", 1152722102
    ],
  "Refbase:=", "Q",
  "NumParts:=", 1,
  "Terminal:=", ["collector", "collector", "A", false, 6, 0, "",
  "Electrical"],
  "Terminal:=", ["base", "base", "A", false, 7, 0, "", "Electrical"],
  "Terminal:=", ["emitter", "emitter", "A", false, 8, 0, "", "Electrical"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ["NAME:Parameters",
    "TextProp:=", ["LabelID","HD", "Property string for netlist ID", "Q@ID"],
    "TextProp:=", ["MOD", "D","Name of model data reference", "required"],
    "VariableProp:=", ["AREA", "D","Emitter area multiplying factor, which
    affects currents, resistances, and capacitances", "1"],
    "VariableProp:=", ["AREAB", "D", "Base AREA","1"],
    "VariableProp:=", ["AREAC", "D", "Collector AREA","1"],
    "VariableProp:=", ["DTEMP", "D","The difference between element and cir-
    cuit temperature (deg Cel)","0"],
    "VariableProp:=", ["M", "D","Multiplier factor to simulate multiple BJTs
    in parallel","1"],
    "ButtonProp:=", ["NexximNetlist", "HD", "", "Q@ID %0 %1 %2 *MOD(@MOD)
    *AREA(AREA=@AREA)"& " *AREAB(AREAB=@AREAB) *AREAC(AREAC=@" &
    "AREAC)*DTEMP(DTEMP=@DTEMP) *M(M=@M)","Q@ID %0 %1 %2 *MOD(@MOD) " &
    "*AREA(AREA=@AREA) *AREAB(AREAB=@AREAB) *AREAC(AREAC=@" &"AREAC) *DTEMP
    (DTEMP=@DTEMP) *M(M=@M)", 1,"ButtonPropClientData:=", []],
    "TextProp:=", ["ModelName", "HD", "", "Q"]
    ]
  ])
```

Python Example

```
name2 = oComponentManager.Edit ("MyComponent",
  ["NAME:MyOtherComponent",
    "Info:=", ["Type:=", 4294901767,
    "NumTerminals:=", 2,
    "DataSource:=", "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ModifiedOn:=", 1071096503,
    "Manufacturer:=", "Ansys",
    "Symbol:=", "bendo",
    "Footprint:=", "BENDO",
    "Description:=", "",
    "InfoTopic:=", "",
    "InfoHelpFile:=", "",
    "IconFile:=", "",
    "LibraryName:=", "",
    "OriginalLocation:=", "Project",
    "Author:=", "",
    "OriginalAuthor:=", "",
    "CreationDate:= ", 1147460679
    ],
    "Refbase:=", "U",
    "NumParts:=", 1,
    "OriginalComponent:=", "",
    "Terminal:=", ["n1", "n1", "A", false, 0, 0, "",  "Electrical"],
    "Terminal:=", ["n2", "n2", "A",  false, 1, 0, "", Electrical"],
    ["NAME:Parameters",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "MenuProp:=", ["CoSimulator", "D","", "Default, Custom, Netlist", 0],
      "ButtonProp:=", ["CosimDefinition",  "D", "", "", "Edit", 0, "But-
      tonPropClientData:=", []]
    ],
    ["NAME:CosimDefinitions",
      ["NAME:CosimDefinition",
        "CosimulatorType:=", 0,
        "CosimDefName:=", "HFSS3D",
        "IsDefinition:=", true,
        "CosimStackup:=", "Layout stackup",
         "CosimDmbedRatio:=", 3],
      ["NAME:CosimDefinition",
        "CosimulatorType:=", 1,
        "CosimDefName:=", "",
        "IsDefinition:=", true,
        "ExportAsNport:=", 0,
        "UsePjt:=", 0],
      ["NAME:CosimDefinition",
        "CosimulatorType:=", 2,
        "CosimDefName:=", "Custom",
        "IsDefinition:=", true, "DefinitionCompName:=", ""],
       ["NAME:CosimDefinition",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"CosimulatorType:=", 3, "CosimDefName:=", "Netlist", "IsDefinition:=", true, "NetlistString:=", ""]]])

EditDataset

Modifies a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Edit**.
Parameters	Name / Type / Description <*OriginalName*> / String / Name of the original dataset. <*DatasetDataArray*> / Array / Data for the modified dataset.
Return Value	None.

Python Syntax	EditDataset (<*OriginalName*> <*DatasetDataArray*>)
Python Example	oProject.EditDataset ("ds1" ["NAME:ds2", ["NAME:Coordinates", ["NAME:Coordinate",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "X:=", 1, "Y:=", 2
      ],
      [
        "NAME:Coordinate",
        "X:=", 3, "Y:=", 4
      ]
    ]
]
)
oDesign.EditDataset ("ds1"
["NAME:ds2",
  ["NAME:Coordinates",
    [
      "NAME:Coordinate",
      "X:=", 1, "Y:=", 2
    ],
    [
      "NAME:Coordinate",
      "X:=", 3, "Y:=", 4
    ]
  ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

EditMaterial

Modifies an existing material.

UI Access	**View/Edit Materials** command in the material editor		
Parameters	Name	Type	Description
	<*OriginalName*>	String	Name of the material before editing.
	<*MatProperties*>	Array	Structured array containing material properties: ["NAME:<New material name>", "CoordinateSystemType:=", <string>, "BulkOrSurfaceType:=" , <integer>, ["NAME:PhysicsTypes", "set:=" , <array containing string physics types>], <Optional ModifierDataArray>, "permeability:=" , <string containing value>, "conductivity:=" , <string containing value>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		"thermal_conductivity:=", <string containing value>, "mass_density:=" , <string containing value>, "specific_heat:=" , <string containing value>, "youngs_modulus:=" , <string containing value>, "poissons_ratio:=" , <string containing value>, "thermal_expansion_coefficient:=", <string containing value>]
<ModifierDataArray>	Array	Optional structured array containing thermal or spatial modifiers: ["NAME:ModifierData", ["NAME:<ThermalModifierData or SpatialModifierData>", "modifier_data:=" , <"thermal_modifier_data" or "spatial_modifier_data">, ["NAME:<all_thermal_modifiers or all_spatial_modifiers>", ["NAME:<modifierName>", "Property::=" , <string property being modified>, "Index::=" , <integer>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"prop_modifier:=" , <"thermal_modifier" or "spatial_modifier">, "use_free_form:=" , <Boolean>, "free_form_value:=" , <string modifier value>,]]]]
Return Value	None.		

Python Syntax	EditMaterial (<*OriginalName*>, <*MatProperties*>)

Python Example	**Without Modifiers:**
	`oDefinitionManager.EditMaterial("alumina_92pct",`
	`[`
	`"NAME:alumina_92pct",`
	`"CoordinateSystemType:=" , "Cartesian",`
	`"BulkOrSurfaceType:=" , 1,`
	`[`
	`"NAME:PhysicsTypes",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "set:=" , ["Electromagnetic","Thermal","Structural"]
  ],
  "permittivity:=" , "9.3",
  "dielectric_loss_tangent:=" , "0.008",
  "thermal_conductivity:=" , "26",
  "mass_density:=" , "3720",
  "specific_heat:=" , "790",
  "youngs_modulus:=" , "267000000000",
  "poissons_ratio:=" , "0.26",
  "thermal_expansion_coeffcient:=", "7.2e-006"
]
)
```

With Thermal Modifier:

```
oDefinitionManager.EditMaterial("copper",
  [
  "NAME:copper",
  "CoordinateSystemType:=", "Cartesian",
  "BulkOrSurfaceType:=" , 1,
    [
    "NAME:PhysicsTypes",
    "set:=" , ["Electromagnetic","Thermal","Structural"]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
  [
  "NAME:ModifierData",
    [
    "NAME:ThermalModifierData",
    "modifier_data:=" , "thermal_modifier_data",
      [
      "NAME:all_thermal_modifiers",
        [
        "NAME:one_thermal_modifier",
        "Property::=" , "permittivity",
        "Index::=" , 0,
        "prop_modifier:=" , "thermal_modifier",
        "use_free_form:=" , True,
        "free_form_value:=" , "if(Temp > 1000cel, 1, if(Temp < -273.15cel, 1, 1))"
        ]
      ]
    ]
  ],
  "permeability:=" , "0.999991",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "conductivity:=" , "58000000",
  "thermal_conductivity:=" , "400",
  "mass_density:=" , "8933",
  "specific_heat:=" , "385",
  "youngs_modulus:=" , "120000000000",
  "poissons_ratio:=" , "0.38",
  "thermal_expansion_coefficient:=" , "1.77e-05"
])
```

Transient Solve, Non-linear Drude Data Plasma

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Drude_plasma_parameters_r231")
oDefinitionManager = oProject.GetDefinitionManager()
oDefinitionManager.EditMaterial("Drude",
  [
    "NAME:Drude",
    "CoordinateSystemType:=", "Cartesian",
    "BulkOrSurfaceType:="   , 1,
    [
      "NAME:PhysicsTypes",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "set:="                 , ["Electromagnetic"]
    ],
    [
        "NAME:AttachedData",
        [
            "NAME:MatNonLinearDrudeFreqDepData",
            "property_data:="     , "nonlinear_drude_data",
            "EpsilonInfinity:="    , "1",
            "PlasmaFrequency:="    , "4.62348462366278GHz",
            "CollisionFrequency:="  , "0.00054491190162662GHz",
            "FieldBreakdown:="     , "10000V_per_meter",
            "PlasmaMaintainFrequency:=", "2.31174231183139GHz",
            "NeutralDensity:="     , 2.65164580488373E+20,
            "ElectronDensity:="    , 2.65164580488373E+17,
            "CollisionRateConstant:=", 2.05499505485618E-15
        ]
    ]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit [padstack manager]

Use: Edit an existing padstack.

Command: Tools > Edit Configured Libraries > Padstacks > Edit Padstack

Syntax: Edit <PadstackName>,

```
Array("NAME:<NewPadstackName>",
"ModTime:=", <ModifiedOnInfo>,
"Library:=", "", // name of the library
"LibLocation:=", "Project", // location of the named library
Array("NAME:psd",
"nam:= ", <PadstackName>,
"lib:=", "", // name of the library
"mat:=", "", // hole plating material
"plt:=", "0", // percent of hole's radius filled by plating
Array("NAME:pds",
<LayerGeometryArray>,
<LayerGeometryArray....),
"hle:=", <PadInfo>
"hRg:=", <HoleRange>,
"sbsh:=", <SolderballShape>,
"sbpl:=", <SolderballPlacement>,
"sbr:=", <string>, // solderball diameter, real with units
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"sb2:=", <string>, // solderball mid diameter, real with units

"sbn:=", <string>), // name of solderball material

"ppl:=", <PadPortLayerArray>)

Return Value: <string> // composite name of the padstack

// If the name requested conflicts with the name of an existing

// padstack, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <PadstackName>:

<string> // composite name of padstack to edit

<NewPadstackName>:

<string> // new simple name for padstack

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed

since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<LayerGeometryArray>:

Array("Name:lgm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential

information of ANSYS, Inc. and its subsidiaries and affiliates.

"lay:=", <string>, // definition layer name

"id:=", <int>, // definition layer id

"pad:=", <PadInfo>, // pad

"ant:=", <PadInfo>, // antipad

"thm:=", <PadInfo>, // themal pad

"X:=", <string>, // pad x connection, real with units

"Y:=", <string>, // pad y connection, real with units

"dir:=", <DirectionString>) // pad connection direction

<PadInfo>:

Array("shp:=", <PadShape>,

"Szs:=", <DimensionArray>,

"X:=", <string>, // x offset, real with units

"Y:=", <string>, // y offset, real with units

"R:=", <string>) // rotation, real with units

<PadShape>:

<string> one of these choices

"No" // no pad

"Cir" // Circle

"Sq" // Square

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Rct" // Rectangle

"Ov" // Oval

"Blt" // Bullet

"Ply" // Polygons

"R45" // Round 45 thermal

"R90" // Round 90 thermal

"S45" // Square 45 thermal

"S90" // Square 90 thermal

<DimensionArray>:

Array(<string>, ...) // each string is a real with units for one of the

// dimensions of the shape

<DirectionString>:

<string> one of these choices

"No" // no direction

"Any" // any direction

"0" // 0 degrees

"45" // 45 degrees

"90" // 90 degrees

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"135" // 135 degrees

"180" // 180 degrees

"225" // 225 degrees

"270" // 270 degrees

"315" // 315 degrees

<HoleRange>:

<string> one of these choices

"Thr" // through all layout layers

"Beg" // from upper pad layer to lowest layout layer

"End" // from upper layout layer to lowest pad layer

"UTL" // from upper pad layer to lowest pad layer

<SolderballShape>:

<string> one of these choices

"None" // no solderball

"Cyl" // cylinder solderball

"Sph" // spheroid solderball

<SolderballPlacement>:

<string> one of these choices

"abv" // above padstack

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"blw" // below padstack

<PadPortLayerArray>:

Array(<int>, <int>,....) where each int is a layer id

ExportDataset

Exports a dataset to a named file. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Export**.
Parameters	Name | Type | Description *<datasetFileFullPath>* | String | The full path to the file.
Return Value	None.

Python Syntax	ExportDataset (*<datasetFileFullPath>*)
Python Example	`oProject.ExportDataset('e:/tmp/dsdata.txt')` `oDesign.ExportDataset('e:/tmp/dsdata.txt')`

Export [footprint manager]

Export a footprint to a library.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Tools** > **Edit Configured Libraries** > **Footprints** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the footprint to <*FootprintName*> – String that defines the composite name of the footprint to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are "`Project`", "`PersonalLib`", or "`UserLib`".
Return Value	None		

Python Syntax	Export(*["NAME:<LibraryName>"*, <*FootprintName*>*]*, <*LibraryLocation*>)
Python Example	`oFootprintManager.Export(["NAME:mylib", "Distributed Footprints:BPAD"], "PersonalLib")`

ExportMaterial

Exports a local material to a library.

UI Access	**Export to Library** command in the material editor.		
Parameters	Name	Type	Description
	<*ExportData*>	Array	["NAME:<LibraryName>", <MaterialName>, <MaterialName>, ...]
	<*LibraryName*>	String	Name of the exported library.
	<*MaterialName*>	String	Name of the material to be exported.
	<*LibraryLocation*>	String	Location to save the library. Only "PersonalLib" and "UserLib" are

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	allowed.
Return Value	None.

Python Syntax	ExportMaterial (<*ExportData*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.ExportMaterial (["NAME:mylib", "Material1", "Material2", "Material3"], "PersonalLib")`

Export [padstack manager]

Exports a padstack to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the padstack to <*PadstackName*> – String that defines the simple name of the padstack to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are `"Project"`, `"PersonalLib"`, or `"UserLib"`.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Export(*[Name:*<*LibraryName*>, <*PadstackName*>], <*LibraryLocation*>)
Python Example	`oPadstackManager.Export(["NAME:mylib", "myPadstack"], "PersonalLib")`

ExportScript

Exports a script to a library in the script definition manager.

UI Access	**None**		
Parameters	Name	Type	Description
	<*ScriptArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the script to <*ScriptName*> – String that defines the name of the script to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are `"Project"`, `"PersonalLib"`, or `"UserLib"`.
Return Value	None		

Python Syntax	ExportScript(<*ScriptArray*>,<*LibraryLocation*>)
Python Example	`oProject.ExportScript(["NAME:mylib", "myscript"], "PersonalLib")`

Export [symbol manager]

Exports symbol(s) to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Symbols** > **Export to Library**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*NameArray*></td><td>Array</td><td>Includes the following: <*LibraryName*> – String that defines the name of the library to export the symbol to <*SymbolName*> – String that defines the composite name of the symbol to export</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location of the library specified with <LibraryName>; options are "Project", "PersonalLib", or "UserLib".</td></tr></table>
Return Value	None

Python Syntax	Export([<*LibraryName*>, <*SymbolName*>*]*, <*LibraryLocation*>)

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`oSymbolManager.Export(["NAME Nexxim Circuit Elements\Distributed\Distributed:bendo:mylib", "mySymbol"], "PersonalLib")`

GetProjectMaterialNames

Returns the material names belonging to an active project.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td>None</td><td></td><td></td></tr></table>
Return Value	String names of the materials in the active project.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetProjectMaterialNames()

Python Example	`oProject = oDesktop.GetActiveProject()`
	`oDefinitionManager = oProject.GetDefinitionManager()`
	`materials = oDefinitionManager.GetProjectMaterialNames()`
	`AddWarningMessage(str(materials))`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

> **Tip:** Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

UI Access	N/A
Parameters	See table below

Name	Type	Description
<*PropTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<PropServer>	String	An object identifier, generally returned from another script method, such as `Comp‡nst@R;2;3`
	<PropName>	String	Name of the property.
Return Value	String value of the property.		

Python Syntax	GetPropertyValue (*<PropTab>*, *<PropServer>*, *<PropName>*)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`
	`...`

ImportDataset

Imports a dataset from a named file. This can be executed by the oProject, or oDesign variables. The name of the dataset is file-name+index number (e.g., dsdata1) unless the filename ends with a trailing number. When there is a trailing number at the end, we

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

will remove the number and use first unused index. Alternatively, the name of the dataset can be explicitly defined by providing a string as an optional second argument.

UI Access	**Project** > **Datasets** > **Import**.		
Parameters	Name	Type	Description
	<datasetFileFullPath>	String	The full path to the file containing the dataset values. *.tab files recommended (see note below).
	<optionalDatasetName>	String	*Optional*. User-defined dataset name.
Return Value	None.		

Python Syntax	ImportDataset (*<datasetFileFullPath>*,*<optionalDatasetName>*)

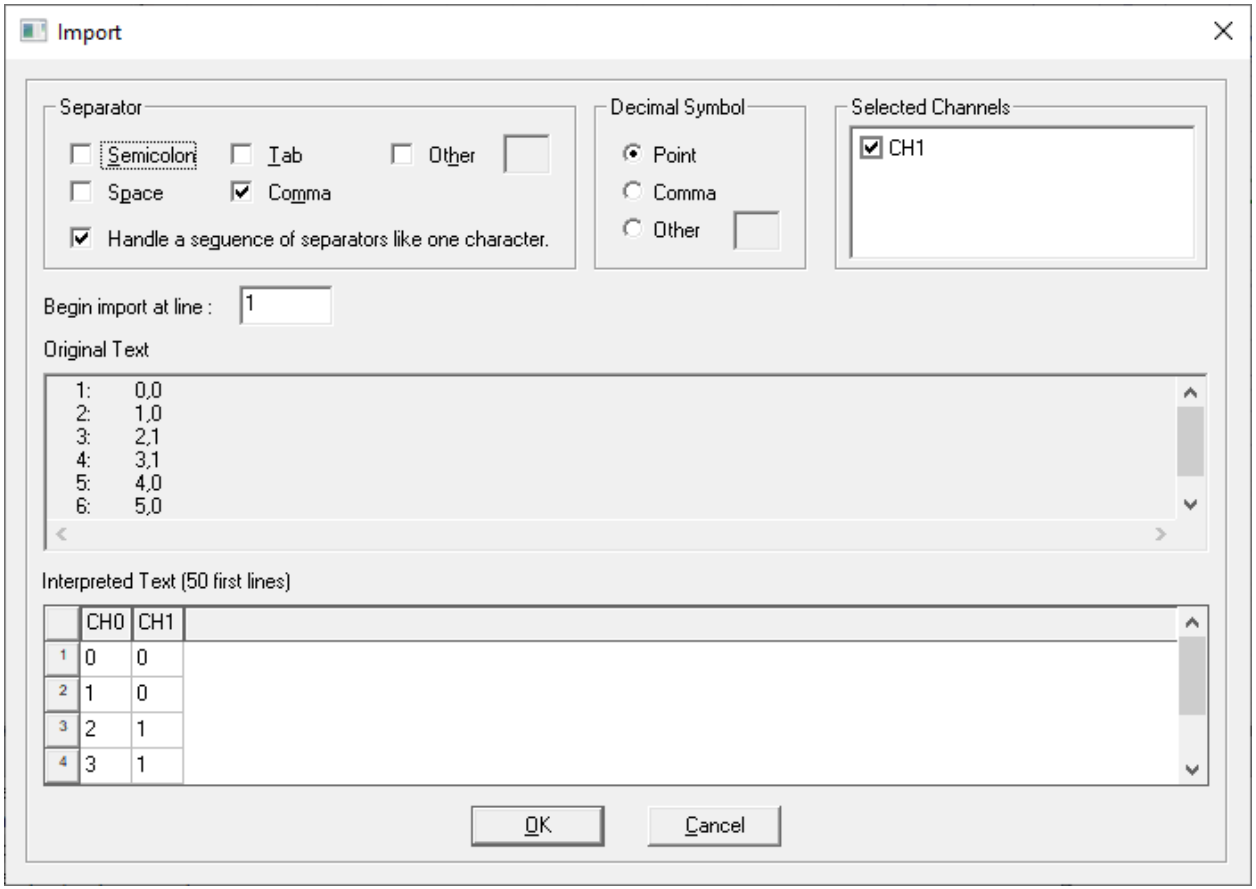
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

Note About File Types:

Tab-delimited or space-delimited files with the extension *.tab are the recommended file type. When using ImportDataset at the Design level, *.tab is the only file type supported.

At the Project level, other file types are supported (for example, *.csv). However, after calling the command, you must configure the file import format manually through the Electronics Desktop GUI by selecting **Project** > **Datasets** and clicking **Import**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Remove [component manager]

Removes a component from a library.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Tools** > **Edit Configured Libraries** > **Components** > **Remove Component**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ComponentName*></td><td>String</td><td>Name of the component to remove</td></tr><tr><td><*IsProjectComponent*></td><td>Boolean</td><td>True if it is a project component</td></tr><tr><td><*LibraryName*></td><td>String</td><td>Name of the library</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".</td></tr></table>
Return Value	None

Python Syntax	Remove (<ComponentName>, <IsProjectComponent>, <LibraryName>, <LibraryLocation>)
Python Example	`oComponentManager.Remove ("Simplorer Circuit Elements\BJTs:Level01_NPN", _ true, "Project")`

Remove [footprint manager]

Removes a footprint from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Footprints** > **Remove Footprint**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*FootprintName*></td><td>String</td><td>Composite name of the footprint to remove</td></tr><tr><td><*IsProjectFootprint*></td><td>Boolean</td><td>True if it is a project footprint</td></tr><tr><td><*LibraryName*></td><td>String</td><td>Name of the library</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	Remove (<*FootprintName*>, <*IsProjectFootprint*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oFootprintManager.Remove("BPAD", false, "MyLib", "PersonalLib")`

RemoveMaterial

Removes a material from a library.

UI Access	**Remove Material(s)** command in the material editor
Parameters	

Name	Type	Description
<*MaterialName*>	String	Name of the material to be removed.
<*IsProjectMaterial*>	Boolean	If True, assumes the material is a project material. The last two parameters will be ignored. If False, the material is not a project material.
<*LibraryName*>	String	Name of the user or personal library where the material resides.
<*LibraryLocation*>	String	Location of library. Valid options:"UserLib" or "PersonalLib".

Return Value	None.

Python Syntax	RemoveMaterial (<*MaterialName*>, <*IsProjectMaterial*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveMaterial ([` `"Material1", false, "mo0907","UserLib"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Remove [padstack manager]

Removes a padstack from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Remove Padstacks**		
Parameters	Name	Type	Description
	<*PadstackName*>	String	Simple name of the padstack to remove
	<*IsProjectPadstack*>	Boolean	True if it is a project padstack
	<*LibraryName*>	String	Name of the library
	<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".
Return Value	None		

Python Syntax	Remove (<PadstackName>, <IsProjectPadstack>, <LibraryName>, <LibraryLocation>)
Python Example	`oPadstackManager.Remove ("Polygon SMT", false, "MyLib", "PersonalLib")`

RemoveScript

Removes a script in the script definition manager.

UI Access	**None**		
Parameters	Name	Type	Description
	<*ScriptName*>	String	Simple name of the script to remove
	<*IsProjectScript*>	Boolean	True if it is a project script
	<*LibraryName*>	String	Name of the library

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".
Return Value	None		

Python Syntax	RemoveScript (<*ScriptName*>,<*IsProjectScript*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveScript(` `"myscript", true, "Local", "Project")`

Remove [symbol manager]

Removes a symbol from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Symbols** > **Remove Symbol**		
Parameters	Name	Type	Description
	<*SymbolName*>	String	Composite name of the symbol to remove
	<*IsProjectSymbol*>	Boolean	True if it is a project symbol
	<*LibraryName*>	String	Name of the library
	<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".
Return Value	None		

Python Syntax	Remove (<SymbolName>, <*IsProjectSymbol*>, <*LibraryName*>, <*LibraryLocation*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oSymbolManager.Remove("Nexxim Circuit Elements\Distributed\Distributed:bendo", true, "Project")`

RemoveUnusedDefinitions

Removes any unused project definitions.

UI Access	**Tools** > **Project Tools** > **Remove Unused Definitions**.
Parameters	Name | Type | Description <*Definitions*> | Array | Definitions to be removed, such as materials and surface materials.
Return Value	None.

Python Syntax	RemoveUnusedDefinitions(<*Definitions*>)

Python Example	`oProject.RemoveUnusedDefinitions(`
	`  [`
	`  [`
	`  "NAME:Materials",`
	`  "Al-Extruded"`
	`  ],`
	`  [`
	`  "NAME:SurfaceMaterials",`
	`  "Steel-oxidised-surface"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]])

SetPropertyValue

Sets the value of a single property belonging to a specific PropServer and PropTab. This function is available with the Project, Design or Editor objects, including definition editors.This is not supported for properties of the following types: ButtonProp, PointProp, V3DPointProp, and VPointProp. Only the ChangeProperty command can be used to modify these properties.

Use the script recording feature and edit a property, and then view the resulting script entry or use GetPropertyValue for the desired property to see the expected format.

UI Access	N/A		
Parameters	Name	Type	Description
	<propTab>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities")

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Signals ("Signals")
	<propServer>	String	An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3
	<propName>	String	Name of the property.
	<propValue>	String	The value for the property
Return Value	None.		

Python Syntax	SetPropertyValue(*<propTab>*, *<propServer>*, *<propName>*, *<propValue>*)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

UpdateDefFromBlock

Updates a material definition from block text (same definition format as would be contained in the material library file) by library type (using definition folder name). This scripting command directly supports the .AMAT (or .ASURF) definition formats.

UI Access	N/A		
Parameters	Name	Type	Description
	<targetDefName>	String	Name of the target definition, i.e. the name of the material to update.
	<defBlock>	String	Text of the new material definition in block format (string); this block could use a new definition name, which will cause a rename as part of the update.
	<defFolderName>	String	Library type (by definition, folder name).
	<newTimeStamp>	String	New timestamp string (time_t as integer, number of seconds since 1/1/1970 12:00am), default is current time.
Return Value	A property scripting object for the definition.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	UpdateDefFromBlock(<*targetDefName*>, <*defBlock*>, <*defFolderName*>, <*newTimeStamp*>)

Python Example	`oProject = oDesktop.NewProject()`
	`oProject.InsertDesign("HFSS", "HFSSDesign1", "DrivenModal", "")`
	`oDesign = oProject.SetActiveDesign("HFSSDesign1")`
	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oDefinitionManager = oProject.GetDefinitionManager()`
	`defBlock = "$begin 'vacuum2' $begin 'AttachedData' $begin 'MatAppearanceData' property_data-='appearance_data' Red=230 Green=230 Blue=230 Transparency=0.95 $end 'MatAppearanceData' $end 'AttachedData' simple('permittivity', 1) ModTime=1499970477 $end 'vacuum2'"`
	`added = oDefinitionManager.AddDefinitionFromBlock(defBlock, "Materials", "10101010", True)`
	`addedName = ''`
	`  if isinstance(added, basestring):`
	`    addedName = added`
	`  elif isinstance(added, list):`
	`    addedName = added[0]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
else:


   addedName = added.GetName().replace("Materials:", "")

AddInfoMessage(os.path.basename(__file__) + " result: " + addedName)

materialNameInQuotes = "\"" + addedName + "\""

# rename vacuum2 to vacuum3

newDefBlock = "$begin 'vacuum3' $begin 'AttachedData' $begin 'MatAppearanceData' property_
data='appearance_data' Red=230 Green=230 Blue=230 Transparency=0.95 $end 'MatAppearanceData'
$end 'AttachedData' simple('permittivity', 1) ModTime=1499970477 $end 'vacuum3'"

updatedObj = UpdateDefFromBlock(addedName, newDefBlock, "Materials")
```

UpdateDefFromBlockEx

Updates a material definition from block text (same definition format as would be contained in the material library file) by library type (using definition folder name). This scripting command directly supports the .AMAT (or .ASURF) definition formats.This resembles the UpdateDefFromBlock command, except UpdateDefFromBlockEx also allows for:

- updating a definition using a block where the name has changed without triggering a rename of the definition being updated
- updating multiple definitions from the same block
- renaming a definition to something other than the name in the block

UI Access	N/A		
Parameters	Name	Type	Description
	<targetDefName>	String	Name of the target definition, i.e. the name of the material to update. Name of the target definition, i.e. the

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			name of the material to update (string). Name of the target definition, i.e. the name of the material to update (string).		
	<*desiredDefName*>	String	New name to use for a rename. If this is the same as *targetDefName*, it means the definition is not being renamed. If this is blank, the block name will be used for this purpose instead.	New name to use for a rename. If this is the same as targetDefName, it means the definition is not being renamed. If this is blank, the block name will be used for this purpose instead.	New name to use for a rename. If this is the same as targetDefName, it means the definition is not being renamed. If this is blank, the block name will be used for this purpose instead.
	<*defBlock*>	String	Text of the updated material definition in block form. (Same form as the defBlock input to the AddDefinitionFromBlock and UpdateDefFromBlock commands, for example.) Note that, unlike in the UpdateDefFromBlock command, the name of the definition in the block can be ignored, provided that desiredDefName is set.		
	<*defFolderName*>	String	Library type (by definition, folder name).		
	<*newTimeStamp*>	String	New timestamp string (time_t as integer, number of seconds since 1/1/1970 12:00am), default is current		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			time.
	<*continueIfConflict*>	Boolean	A flag for whether to continue processing if there is some sort of conflict. A conflict only comes from the target definition being in use, such as a material assigned to a particular part, and if trying to rename the target definition. If this argument is True, and there's a rename conflict, the tactic is to add/update the definition with the new name and leave the target definition untouched. If this argument is False, and there's a rename conflict, the command results in an exception.

Return Value	A property scripting object (extended definition object) for the definition. (In the event of a failure, the command throws an exception and there will be no result.)

Python Syntax	UpdateDefFromBlockEx(<*targetDefName*>, <*desiredDefName*>,<*defBlock*>, <*defFolderName*>, <*newTimeStamp*>, <*continueIfConflict*>)

Python Example	`# Note that the material block name will be ignored in this case`
	`defBlock = "$begin 'unused_name' $begin 'AttachedData' $begin \`
	`'MatAppearanceData' property_data='appearance_data' \`
	`Red=255 Green=0 Blue=0 Transparency=0.19 $end \`
	`'MatAppearanceData' $end 'AttachedData' \`
	`simple('permittivity', 1.9) ModTime=1509090909 \`
	`$end 'unused_name'"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	try:
	  updated = oDefinitionManager.UpdateDefFromBlockEx("red_material",
	  "red_material",
	  defBlock,
	  "Materials",
	  "1609090909")
	  updatedName = updated.GetName().replace("Materials:", "")
	  AddInfoMessage("Result: " + updatedName)
	except:
	  AddErrorMessage("Unexpected error in UpdateDefFromBlockEx")

UpdateDefinitions

Updates all definitions. The **Messages** window reports when definitions are updated, or warns when definitions cannot be found.

UI Access	**Tools** > **Project Tools** > **Update Definitions**. Click **Select All**, then **Update**.
Parameters	None.
Return Value	None.

Python Syntax	UpdateDefinitions()
Python Example	oProject.UpdateDefinitions()

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Component Manager Script Commands

The component manager provides access to components in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
oComponentManager = oDefinitionManager.GetManager("Component")
```

The topics for this section include:

Add

AddNPortData

AddSolverOnDemandModel

Edit

EditSolverOnDemandModel

EditWithComps

Export

GetNPortData

GetSolverOnDemandData

GetSolverOnDemandModelList

Remove

RemoveSolverOnDemandMode

UpdateDynamicLink

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Add [component manager]

Add creates a new component in the component library.

<table>
<tr><td>UI Access</td><td colspan="3">Tools > Edit Libraries > Component > Add</td></tr>
<tr><td rowspan="25">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><NAME></td><td>string</td><td>Component name.</td></tr>
<tr><td><Info></td><td></td><td>Lists the component information described below.</td></tr>
<tr><td><Type></td><td>int</td><td>Component type.</td></tr>
<tr><td><NumTerminals></td><td>int</td><td>Number of terminals.</td></tr>
<tr><td><DataSource></td><td>string</td><td>Data source specified.</td></tr>
<tr><td><ModifiedOn></td><td>int</td><td>Date of last change to the component.</td></tr>
<tr><td><Manufacturer></td><td>string</td><td>Component manufacturer.</td></tr>
<tr><td><Symbol></td><td>string</td><td>Symbol used to represent the component.</td></tr>
<tr><td><Component></td><td>string</td><td>Component</td></tr>
<tr><td><ModelNames></td><td>string</td><td>Component model name</td></tr>
<tr><td><Footprint></td><td>string</td><td>Component footprint.</td></tr>
<tr><td><Description></td><td>string</td><td>Component description.</td></tr>
<tr><td><InfoTopic></td><td>string</td><td>Component information topic file.</td></tr>
<tr><td><InfoHelpFile></td><td>string</td><td>Path to the help file.</td></tr>
<tr><td><IconFile></td><td>string</td><td>Path to the component icon.</td></tr>
<tr><td><Library></td><td>string</td><td>Component library.</td></tr>
<tr><td><OriginalLocation></td><td>string</td><td>Original location of the component.</td></tr>
<tr><td><Project></td><td>string</td><td>Project of the component.</td></tr>
<tr><td><IEEE></td><td>string</td><td>Component IEEE information.</td></tr>
<tr><td><Author></td><td>string</td><td>Most recent editor of the component.</td></tr>
<tr><td><OriginalAuthor></td><td>string</td><td>Original creator of the component.</td></tr>
<tr><td><CreationDate></td><td>string</td><td>Date of component when first created.</td></tr>
<tr><td><ExampleFile></td><td>string</td><td>Path to component example file.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<HiddenComponent>	int	Number of hidden components.
	<GroupID>	string	Identification tag of the component group.
	<Refbase>	string	Component reference designator.
	<NumParts>	int	Number of parts in the component.
	<ModSinceLib>	bool	True or False
	<CompExtID>	int	External component identification number.
Return Value	None		

Python Syntax	oComponentManager.Add([ComponentInfo])

Python Example	`oComponentManager.Add(`
	`  ["NAME:Component",`
	`  "Info:=", ["Type:=", 0,`
	`  "NumTerminals:=", 0,`
	`  "DataSource:=" , "",`
	`  "ModifiedOn:=" , 1747683313,`
	`  "Manufacturer:=", "","Symbol:=", "Component", "ModelNames:=", "",`
	`  "Footprint:=",`
	`  "Description:=", "",`
	`  "InfoTopic:=", "",`
	`  "InfoHelpFile:=", "",`
	`  "IconFile:=", "",`
	`  "Library:=" , "",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "OriginalLocation:=" , "Project",
        "IEEE:=" , "", "
        Author:=" , "",
        "OriginalAuthor:=" , "",
         "CreationDate:=" , 1747683307,
         "ExampleFile:=" , "",
        "HiddenComponent:=" , 0,
        "CircuitEnv:=" , 0,
        "GroupID:=" , 0
      ],
      "CircuitEnv:=", 0,
      "Refbase:=" , "U",
      "NumParts:=" , 1,
      "ModSinceLib:=" , True,
      "CompExtID:=" , 1
    ])
```

AddNPortData [component manager]

Adds a component using the specified data.

UI Access	**Project** Menu > **Add Model** > **Add Nport Model**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<ComponentDataName>	String	Name of component to be created
<ComponentDataType>	String	Should be defined as "NportData"
<name>	String	Name of the item
<filename>	String	Path to the file that has the desired data
<numberofports>	Integer	
<filelocation>	String	Possible values are "UsePath", "PersonalLib", "UserLib", "SysLib", or "Project".
<domain>	String	Possible values are "time" or "frequency"
<datamode>	String	Possible values are "EnterData", "Import", or "Link"
<devicename>	String	
<ImpedanceTab>	Boolean	
<NoiseDataTab>	Boolean	
<DCBehaviorTab>	Boolean	
<SolutionName>	String	
<dcInfo>	String	Possible values include "DCOpen", "DCShort", "DCShShort", "DCNone", or "DCEmpty".
<displayformat>	String	Possible values are "MagnitudePhase","RealImaginary", or "DbPhase".
<datatype>	String	Possible values are "SMatrix", "YMatrix", or "ZMatrix"
<ShowRefPin>	Boolean	
<RefNodeCheckbox>	Boolean	

<ProductOptionsInfo>	Array	The parameters differ by product:
		HFSS 3D Layout - doesn't support interpolation/DC behavior
		`"DCOption:=", -1,`
		`"InterpOption:=", -1,`
		`"ExtrapOption:=",-1,`
		`"DataType:=", 0`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Nexxim

```
"DCOption:=", <NexximDCOption>,
"InterpOption:=", <NexximInterpOption>,
"ExtrapOption:=", <NexximExtrapOption>,
"DataType:=", 2

  <NexximDCOption>:

  <int> // 0 : Zero Padding
  // 1 : Same as last point
  // 2 : Linear extrapolation from last 2 points
  // 3 : Constant magnitude, linear phase extra-
  polation
  // 4 : Leave all signal lines open circuited
  // 5 : Short all signal lines together
  // 6 : Short all signal lines to ground


  <NexximInterpOption>:

  <int> // 0 : Step
  // 1 : Linear


  <NexximExtrapOption>:

  <int> // 0 : Zero padding
  // 1 : Same as last point
  // 2 : Linear extrapolation from last 2 points
  // 3 : Constant magnitude, linear phase extra-
  polation
```

Circuit

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<CircuitDCOption>: <int> // 0 : Leave all signal lines open circuited // 1 : Short all signal lines together // 2 : Short all signal lines to ground // 3 : Extrapolate from data provided (not recommended) <CircuitInterpOption>: <int> // 0 : Linear // 1 : Cubic spline // 2 : Rational polynomial <CircuitExtrapOption>: <int> // 0 : Same as interpolation // 1 : Zero padding // 2 : Same as last point
Return Value	String that includes the composite name of the component. If the name requested conflicts with the name of an existingcomponent, the requested name is altered to be unique. The name returned reflects any change made to be unique.		

Python Syntax	AddNPortData ([<ComponentDataName>, <ComponentDataType>, <name>, <filename>, <numberofports>, <filelocation>, <LocationType>, <domain>, <datamode>, <devicename>,<ImpedanceTab>,<NoiseDataTab>, <DCBehaviorTab>, <SolutionName>, <displayformat>, <datatype>, <ShowRefPin>, <RefNodeCheckbox>, <ProductOptionsInfo>])
Python Example	oComponentManager.AddNPortData([NAME:ComponentData", "ComponentDataType:=", "NPortData",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"name:=", "NportData",
"filename:=", "",
"numberofports:=", 2,
"filelocation:=", "UsePath",
"domain:=", "frequency",
"datamode:=", "Import",
"devicename:=", "",
"ImpedenceTab:=", true,
"NoiseDataTab:=", true,
"DCBehaviorTab:=", true,
"SolutionName:=", "",
"displayformat:=", "MagnitudePhase",
"datatype:=", "SMatrix",
"ShowRefPin:=", true,
"RefNodeCheckbox:=", false,
"DCOption:=", 3,
"InterpOption:=", 1,
"ExtrapOption:=", 3,
"DataType:=", 2,
"DCOption:=", 3,
"InterpOption:=", 1,
"ExtrapOption:=", 3,
"DataType:=", 2
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Add (for Solver On Demand Model)

The Add command is a general command that can be used in several methods. This topic demonstrates how it can be used to add a Solver On Demand model definition to an existing component. It returns the name of the Solver On Demand model added.

<table>
<tr><td>UI Access</td><td>Right click on a component in the Project Manager tree, and select Edit Component, then select the Solver-On-Demand tab.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ModelName></td><td>Array</td><td>Model to add for cosimulation, with its associated properties</td></tr>
<tr><td><CosimDefinitions></td><td>Array</td><td>Array that defines the cosimulation parameters</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Returns the name of the model added.</td></tr>
</table>

To add a model to the Solver On Demand functionality, the script must do the following:

1. Open the project that includes the design you want to add.
2. Add the design from that project to the current project.
3. Add that design to the Component Manager.
4. Edit the component that you want to set up for Solid On Demand to couple it with the new design.

The example that follows illustrates all these steps.

```
oDesktop.OpenProject("C:/Program Files/ANSYS Inc/v261/AnsysEM/Examples/HFSS/Transmission Lines/-
coplanar_wg_w_ground.aedt")
oProject = oDesktop.SetActiveProject("Interposer_SI")
oDefinitionManager = oProject.GetDefinitionManager()
oModelManager = oDefinitionManager.GetManager("Model")
oModelManager.Add(
  [
    "NAME:Co-Planar Waveguide w Ground",
    "Name:=" , "Co-Planar Waveguide w Ground",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"ModTime:=" , 1765993379,
"Library:=" , "",
"LibLocation:=" , "Project",
"ModelType:=" , "hfss",
"Description:=" , "",
"ImageFile:=" , "C:/Program Files/ANSYS Inc/v261/AnsysEM/Examples/HFSS 3D Layout/HFSS IC/In-
terposer_SI.aedtresults/4E2C35440176599340310.gif",
"SymbolPinConfiguration:=", 0,
[
  "NAME:PortInfoBlk"
],
[
  "NAME:PortOrderBlk"
],
"DesignName:=" , "Co-Planar Waveguide w Ground",
"SolutionName:=" , "3GHz_Opt : LastAdaptive",
"NewToOldMap:=" , [],
"OldToNewMap:=" , [],
"PinNames:=" , ["1a","2a"],
[
  "NAME:DesignerCustomization",
  "DCOption:=" , 0,
  "InterpOption:=" , 0,
  "ExtrapOption:=" , 1,
  "Convolution:=" , 0,
  "Passivity:=" , 0,
  "Reciprocal:=" , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ModelOption:=" , "",
    "DataType:=" , 1
  ],
  [
    "NAME:NexximCustomization",
    "DCOption:=" , 3,
    "InterpOption:=" , 1,
    "ExtrapOption:=" , 3,
    "Convolution:=" , 0,
    "Passivity:=" , 0,
    "Reciprocal:=" , False,
    "ModelOption:=" , "",
    "DataType:=" , 2
  ],
  [
    "NAME:HSpiceCustomization",
    "DCOption:=" , 1,
    "InterpOption:=" , 2,
    "ExtrapOption:=" , 3,
    "Convolution:=" , 0,
    "Passivity:=" , 0,
    "Reciprocal:=" , False,
    "ModelOption:=" , "",
    "DataType:=" , 3
  ],
  "NoiseModelOption:=" , "External",
  "WB_SystemID:=" , "",
  "IsWBModel:=" , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "filename:=" , "C:/Program Files/ANSYS Inc/v261/AnsysEM/Examples/HFSS/Transmission Lines/-
    coplanar_wg_w_ground.aedt",
    "numberofports:=" , 2,
    "Simulate:=" , False,
    "CloseProject:=" , False,
    "SaveProject:=" , True,
    "InterpY:=" , True,
    "InterpAlg:=" , "auto",
    "IgnoreDepVars:=" , False,
    "Renormalize:=" , False,
    "RenormImpedance:=" , 50
  ])
oComponentManager = oDefinitionManager.GetManager("Component")
oComponentManager.Add(
  [
    "NAME:Co-Planar Waveguide w Ground",
    "Info:=" , [ "Type:=" , 8, "NumTerminals:=" , 2, "DataSource:=" , "", "ModifiedOn:=" ,
    1765993415, "Manufacturer:=" , "", "Symbol:=" , "", "ModelNames:=" , "", "Footprint:=" , "",
     "Description:=" , "", "InfoTopic:=" , "", "InfoHelpFile:=" , "", "IconFile:=" , "hfss.bmp",
     "Library:=" , "", "OriginalLocation:=" , "Project", "IEEE:=" , "", "Author:=" , "", "Ori-
    ginalAuthor:=" , "", "CreationDate:=" , 1765993415, "ExampleFile:=" , "", "Hid-
    denComponent:=" , 0, "CircuitEnv:=" , 0, "GroupID:=" , 0],
    "CircuitEnv:=" , 0,
    "Refbase:=" , "S",
    "NumParts:=" , 1,
    "ModSinceLib:=" , False,
    "Terminal:=" , ["1a","1a","A",False,0,1,"","Electrical","0"],
    "Terminal:=" , ["2a","2a","A",False,1,1,"","Electrical","0"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:Properties",
  "TextProp:=" , ["Owner","RD","","HFSS"]
],
"CompExtID:=" , 5,
[
  "NAME:Parameters",
  "VariableProp:=" , ["S","D","","10mil"],
  "VariableProp:=" , ["W","D","","9mil"],
  "VariableProp:=" , ["ModelLength","D","","2mil"],
  "VariableProp:=" , ["gap","D","","6mil"],
  "VariableProp:=" , ["SubstrateThickness","D","","6mil"],
  "VariableProp:=" , ["ConductorThickness","D","","0.7mil"],
  "VariableProp:=" , ["ModelWidth","D","","120mil"],
  "VariableProp:=" , ["nport_losstan","D","","0"],
  "TextProp:=" , ["ModelName","RD","","FieldSolver"],
  "MenuProp:=" , ["CoSimulator","SD","","Default",0],
  "ButtonProp:=" , ["CosimDefinition","SD","","Edit","Edit",40501, "ButtonPropClientData:=",
  []]
],
[
  "NAME:CosimDefinitions",
  [
    "NAME:CosimDefinition",
    "CosimulatorType:=" , 103,
    "CosimDefName:=" , "Default",
    "IsDefinition:=" , True,
    "Connect:=" , True,
    "ModelDefinitionName:=" , "Co-Planar Waveguide w Ground",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "ShowRefPin2:=" , 2,
          "LenPropName:=" , ""
        ],
        "DefaultCosim:=" , "Default"
      ]
    ])
oComponentManager.Edit("HFSS-IC",
  [
    "NAME:HFSS-IC2",
    "Info:=" , [ "Type:=" , 16, "NumTerminals:=" , 10, "DataSource:=" , "", "ModifiedOn:=" ,
    1765993429, "Manufacturer:=" , "", "Symbol:=" , "HFSS-IC", "ModelNames:=" , "", "Foot-
    print:=" , "_symbolFootprint_a8af25e8_a8dcdf60", "Description:=" , "", "InfoTopic:=" , "",
    "InfoHelpFile:=" , "", "IconFile:=" , "Design.bmp", "Library:=" , "", "OriginalLocation:=" ,
    "Project", "IEEE:=" , "", "Author:=" , "", "OriginalAuthor:=" , "", "CreationDate:=" ,
    1715280111, "ExampleFile:=" , "", "HiddenComponent:=" , 0, "CircuitEnv:=" , 0, "GroupID:=" ,
    0],
    "CircuitEnv:=" , 0,
    "Refbase:=" , "U",
    "NumParts:=" , 1,
    "ModSinceLib:=" , True,
    "Terminal:=" , ["U0.A3","U0.A3","A",False,98,1,"","Electrical","0"],
    "Terminal:=" , ["U0.A1","U0.A1","A",False,99,1,"","Electrical","0"],
    "Terminal:=" , ["U0.A5","U0.A5","A",False,100,1,"","Electrical","0"],
    "Terminal:=" , ["U0.A4","U0.A4","A",False,101,1,"","Electrical","0"],
    "Terminal:=" , ["U0.A2","U0.A2","A",False,102,1,"","Electrical","0"],
    "Terminal:=" , ["U1.A4","U1.A4","A",False,103,1,"","Electrical","0"],
    "Terminal:=" , ["U1.A1","U1.A1","A",False,104,1,"","Electrical","0"],
    "Terminal:=" , ["U1.A3","U1.A3","A",False,105,1,"","Electrical","0"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Terminal:=" , ["U1.A2","U1.A2","A",False,107,1,"","Electrical","0"],
"Terminal:=" , ["U1.A5","U1.A5","A",False,110,1,"","Electrical","0"],
[
  "NAME:Properties",
  "TextProp:=" , ["Representation","SRD","","HFSS-IC"],
  "TextProp:=" , ["Owner","SRD","","HFSS"]
],
"CompExtID:=" , 6,
[
  "NAME:Parameters",
  "VariableProp:=" , ["S","D","","10mil"],
  "VariableProp:=" , ["W","D","","9mil"],
  "VariableProp:=" , ["ModelLength","D","","2mil"],
  "VariableProp:=" , ["gap","D","","6mil"],
  "VariableProp:=" , ["SubstrateThickness","D","","6mil"],
  "VariableProp:=" , ["ConductorThickness","D","","0.7mil"],
  "VariableProp:=" , ["ModelWidth","D","","120mil"],
  "VariableProp:=" , ["nport_losstan","D","","0"],
  "TextProp:=" , ["ModelName","SHD","","FieldSolver"],
  "ButtonProp:=" , ["CosimDefinition","D","","Edit","Edit",40501, "ButtonPropClientData:=",
  []],
  "MenuProp:=" , ["CoSimulator","D","","DefaultNetlist,HFSS Model",0]
],
[
  "NAME:CosimDefinitions",
  [
    "NAME:CosimDefinition",
    "CosimulatorType:=" , 4,
    "CosimDefName:=" , "DefaultNetlist",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "IsDefinition:=" , True,
        "Connect:=" , True,
        "Data:=" , [],
        "GRef:=" , []
      ],
      [
        "NAME:CosimDefinition",
        "CosimulatorType:=" , 103,
        "CosimDefName:=" , "HFSS Model",
        "IsDefinition:=" , True,
        "Connect:=" , True,
        "ModelDefinitionName:=" , "Co-Planar Waveguide w Ground",
        "ShowRefPin2:=" , 2,
        "LenPropName:=" , ""
      ],
      "DefaultCosim:=" , "DefaultNetlist"
    ]
  ])
```

ClearSolutionCache [component manager]

Clears the solution cache for dynamic link component.

UI Access	**Dynamic Link Item RCM** > **Clear Solution Cache** or **Dynamic Link Component in schematic RCM** > **Clear Solution Cache**
Parameters	

Name	Type	Description
<*ComponentName*>	String	Name of the dynamic link component

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	ClearSolutionCache *(<Component Name>)*
Python Example	

Edit [component manager]

Modifies an existing component.

UI Access	**Tools** > **Edit Configured Libraries** > **Components** > **Edit Component**

Parameters

Name	Type	
<*ComponentName*>	String	Name of component to edit
<NewComponentName>	String	Name of the new component created from the edits
<*Info*>	Array	Provides the component info. It includes the following: "Type:=", <TypeInfo>,# see more information below "NumTerminals:=", <int>, "DataSource:=", <string>, "ModifiedOn:=", <ModifiedOnInfo>, "Manufacturer:=", "<string>, "Symbol:=", <string>, "Footprint:=", <string>, "Description:=", <string>, "InfoTopic:=", <string>, "InfoHelpFile:=", <string>, "IconFile:=", <string>, "LibraryName:=", <string>, "OriginalLocation:=", <string>, // Project Location "Author:=", <string>, "OriginalAuthor:=", <string>, "CreationDate:= ", <int>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<RefBase>	String	Reference designator
<NumParts>	Integer	Parts per component
<OriginalComponent>	String	Name of the original component
<Terminal>	Array	

<Parameters>	Array	Optional; defines the parameters of the new component. It can include any or all of the following:
		`"VariableProp:=", <VariableInfo>,`
		`"CheckboxProp:=", <CheckBoxInfo>,`
		`"ButtonProp:=", <ButtonInfo>,`
		`"TextProp:=", <TextInfo>,`
		`"NumberProp:=", <NumberInfo>,`
		`"SeparatorProp:=", <SeparatorInfo>,`
		`"ValueProp:=", <ValueInfo>,`
		`"MenuProp:=", <MenuInfo>`

<Properties>	Array	Optional; defines the properties of the new component. It can include any or all of the following:
		`"CheckboxProp:=", <CheckBoxInfo>,`
		`"TextProp:=", <TextInfo>,`
		`"NumberProp:=", <NumberInfo>,`
		`"SeparatorProp:=",<SeparatorInfo>,`
		`"ValueProp:=", <ValueInfo>,`
		`"MenuProp:=", <MenuInfo>),`
		`"VPointProp:=", <VPointInfo>,`
		`"PointProp:=", <PointInfo>`
		The array structure for each property is defined below.

<Quantities>	Array	`("Quantities","QuantityProp:=", <Quant-ityPropInfo>...)` The <QuantityPropInfo> array is defined below.
<CosimDefInfo>	Array	`("NAME:CosimDefinitions", <CosimDe-`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			fInfo>,<CosimDefInfo>...) The<CosimDefInfo> array is defined below.
Return Value	The composite name of the component. If the name requested conflicts with the name of an existing component, the requested name is altered to be unique. The name returned reflects any change made to be unique.		

<TypeInfo>:

An integer that is the or-ing of bits for each product listed below. The default setting is 0xffffffff (4294967295), which indicates valid for all products. In the component editing dialog box, checking different boxes in the "Specify products for which this component is valid" grid control sets specific flags that correspond to the following hex/decimal settings:

```
Nexxim – 100 binary, 4 decimal, 0x4

SIwaveDeNovo – 1000 binary, 8 decimal, 0x8

Simplorer – 10000 binary, 16 decimal, 0x10

MaxwellCircuit – 100000 binary, 32 decimal, 0x20
```

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<TerminalInfo>:

```
[<string>, // symbol pin
<string> // footprint pin
<string >, // gate name
<bool>, // shared
<int>, // equivalence number
<int>, // what to do if unconnected: flag as error:0, ignore:1
<string>, // description
<Nature>]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<Nature>:
<string> // content varies as follows

Nexxim/Circuit:
"Electrical" // the only choice

Simplorer:
# choices include "Electrical", "Magnetic", "Fluidic", "Translational",
"Translational_V", "Rotational", "Rotational_V",
""Radiant", "Thermal", or <VHDLPackageName>

<VHDLPackageName>:
<string> # in the form <Library>.<Package>
<Library>:
<string> # name of the VHDL library
<Package>:
<string> # name of the VHDL package
```

<VariableInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: number, variable, or expression

<FlagLetters>:
<string> # "D" - has description parameter,
# "RD" - readonly & has description parameter,
# or "RHD" - readonly, hidden, & has description parameter
```

<CheckBoxInfo>:

```
[<string>, # name
<FlagLetters>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<string>, # description
"CB:=", <string>, # optional - script for call back
<bool>] # value: true or false
```

<ButtonInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
<string>, # button title
<string>, # extra text
<ClientID>,
 "ButtonPropClientData:= ", <ClientDataArray>]

<ClientID>:
<int> # specifies Button Prop Client:
# 0 - unknown, "ButtonPropClientData" array will be empty
# 1 - Netlist Prop Client
# 2 - not used
# 3 - File Name Prop Client

<ClientDataArray>:
varies with <ClientID>

<ClientID> is 0 or 1: empty array[]
<ClientID> is 3:
["InternalFormatText:=", "<prefix><RelativePath>"]
<prefix>:
<string> # "<Project>", "<PersonalLib>", "<UserLib>", or "<SysLib>"
<RelativePath>:
<string> # relative path to file from <prefix>
```

<TextInfo>:

```
[<string>, # name
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a text string
```

<NumberInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<real>, # value: a number
<string>] # units
```

<SeparatorInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a text string
```

<ValueInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>] # value: a number, variable or expression
```

<MenuPropInfo>:

```
[<string>, # name
<FlagLetters>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<string>, # description
<string>, # menu choices - separated by commas
<int>] # 0 based index of current menu choice
```

<VPointInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>, # x value: number with length units
<string>] # y value: number with length units
```

<PointInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<real>, # x value
<real>] # y value
```

<QuantityPropInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
<string>, # value
<TypeString>, # string, options are "Across", "Through", or "Free"
<TypeStringDependentInfo>]

<TypeStringDependentInfo>:
"Free" :
<string>, # direction: "In", "Out", "InOut", or "DontCare"
# Following <string> is not present if direction is "DontCare"
<string> # when to calculate: "BeforeAnalogSolver",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
// "BeforeStateGraph", "AfterStateGraph", or "DontCareWhen"
"Across" or "Through"
<int>, # terminal 1
<int> # terminal 2
```

<CosimDefInfo>:

```
["NAME:CosimDefinition",
"CosimulatorType:=", <int>,
"CosimDefName:=", <string> # "HFSS3D", "Circuit", "Custom", or "Netlist"
"IsDefinition:=", <bool>,
final array members]
```

Final array member(s) vary with CosimDefName:

- final array members for HFSS 3D Layout:

  ```
  "CosimStackup:=", <string>,
  "CosimDmbedRatio:=", <int>
  ```

- final array members for Circuit:

  ```
  "ExportAsNport:=", <int>,
  "UsePjt:=", <int>
  ```

- final array member for Custom:

  ```
  "DefinitionCompName:=", <string>
  ```

- final array member for Netlist:

  ```
  "NetlistString:=", <string>
  ```

Python Syntax	Edit (<*ComponentName*>, [<*NewComponentName*>", <*Info*>, <*RefBase*>,<*NumParts*>, <*Terminals*>, <*Parameters*>,<*Properties*>, <*Quantities*>,<*CosimDefInfo*>])
Python Example	`name = oComponentManager.Edit ("Simplorer Circuit Elements\BJTs:Level01_NPN",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:Level01_NPN",
  "Info:=", ["Type:=", 4294901764,
    "NumTerminals:=", 3,
    "DataSource:=", "Ansoft built-in component",
    "ModifiedOn:=", 1152722112,
    "Manufacturer:=", "",
    "Symbol:=", "nexx_bjt_npn",
    "Footprint:=", "",
    "Description:=", "BJT, GP, NPN",
    "InfoTopic:=", "NXBJT1.htm",
    "InfoHelpFile:=", "nexximcomponents.chm",
    "IconFile:=", "bjtsn.bmp",
    "Library:=", "Nexxim Circuit Elements\BJTs",
    "OriginalLocation:=", "SysLibrary ",
    "Author:=", "",
    "OriginalAuthor:=", "",
    "CreationDate:=", 1152722102
  ],
"Refbase:=", "Q",
"NumParts:=", 1,
"Terminal:=", ["collector", "collector", "A", false, 6, 0, "",
"Electrical"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Terminal:=", ["base", "base", "A", false, 7, 0, "", "Electrical"],
  "Terminal:=", ["emitter", "emitter", "A", false, 8, 0, "", "Electrical"],
  ["NAME:Parameters",
    "TextProp:=", ["LabelID","HD", "Property string for netlist ID", "Q@ID"],
    "TextProp:=", ["MOD", "D","Name of model data reference", "required"],
    "VariableProp:=", ["AREA", "D","Emitter area multiplying factor, which
    affects currents, resistances, and capacitances", "1"],
    "VariableProp:=", ["AREAB", "D", "Base AREA","1"],
    "VariableProp:=", ["AREAC", "D", "Collector AREA","1"],
    "VariableProp:=", ["DTEMP", "D","The difference between element and cir-
    cuit temperature (deg Cel)","0"],
    "VariableProp:=", ["M", "D","Multiplier factor to simulate multiple BJTs
    in parallel","1"],
    "ButtonProp:=", ["NexximNetlist", "HD", "", "Q@ID %0 %1 %2 *MOD(@MOD)
    *AREA(AREA=@AREA)"& " *AREAB(AREAB=@AREAB) *AREAC(AREAC=@" &
    "AREAC)*DTEMP(DTEMP=@DTEMP) *M(M=@M)","Q@ID %0 %1 %2 *MOD(@MOD) " &
    "*AREA(AREA=@AREA) *AREAB(AREAB=@AREAB) *AREAC(AREAC=@" &"AREAC) *DTEMP
    (DTEMP=@DTEMP) *M(M=@M)", 1,"ButtonPropClientData:=", []],
    "TextProp:=", ["ModelName", "HD", "", "Q"]
    ]
  ])
```

Python Example

```
name2 = oComponentManager.Edit ("MyComponent",
  ["NAME:MyOtherComponent",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Info:=", ["Type:=", 4294901767,
"NumTerminals:=", 2,
"DataSource:=", "",
"ModifiedOn:=", 1071096503,
"Manufacturer:=", "Ansys",
"Symbol:=", "bendo",
"Footprint:=", "BENDO",
"Description:=", "",
"InfoTopic:=", "",
"InfoHelpFile:=", "",
"IconFile:=", "",
"LibraryName:=", "",
"OriginalLocation:=", "Project",
"Author:=", "",
"OriginalAuthor:=", "",
"CreationDate:= ", 1147460679
],
"Refbase:=", "U",
"NumParts:=", 1,
"OriginalComponent:=", "",
"Terminal:=", ["n1", "n1", "A", false, 0, 0, "",  "Electrical"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Terminal:=", ["n2", "n2", "A",  false, 1, 0, "", Electrical"],
["NAME:Parameters",
  "MenuProp:=", ["CoSimulator", "D","", "Default, Custom, Netlist", 0],
  "ButtonProp:=", ["CosimDefinition",  "D", "", "", "Edit", 0, "But-
  tonPropClientData:=", []]
],
["NAME:CosimDefinitions",
  ["NAME:CosimDefinition",
    "CosimulatorType:=", 0,
    "CosimDefName:=", "HFSS3D",
    "IsDefinition:=", true,
    "CosimStackup:=", "Layout stackup",
     "CosimDmbedRatio:=", 3],
  ["NAME:CosimDefinition",
    "CosimulatorType:=", 1,
    "CosimDefName:=", "",
    "IsDefinition:=", true,
    "ExportAsNport:=", 0,
    "UsePjt:=", 0],
  ["NAME:CosimDefinition",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "CosimulatorType:=", 2,
          "CosimDefName:=", "Custom",
          "IsDefinition:=", true, "DefinitionCompName:=", ""],
        ["NAME:CosimDefinition",
          "CosimulatorType:=", 3,
          "CosimDefName:=", "Netlist",
          "IsDefinition:=", true,  "NetlistString:=", ""]
    ]
  ])
```

EditSolverOnDemandModel

This method looks for a local component of the name passed in, and in this component it looks for an SOD model using the name passed in the second BSTR. It modifies the SOD model using the data in the VARIANT.

UI Access	Not available
Parameters	

Name	Type	Description
<*BSTRName*>	String	BSTR component name
<*BSTRSODName*>	String	BSTR SOD model name
<*VARIANT*>		VARIANT that is the new SOD model data (can include changed name)

Return Value	Returns the name of the model edited.

Python Syntax	EditSolverOnDemandModel (<*BSTRName*>,<*BSTRSODName*>, <*VARIANT*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	

EditWithComps [component manager]

Edit an existing component.

UI Access	Not available		
Parameters	Name	Type	Description
	<*ComponentName*>	String	Composite name of the component being edited
	<*ComponentPropertiesArray*>	Array	Array of component properties
	<*NAME*>	String	New simple name for the component
	<*ModTime*>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
	<*Library*>	String	Library name
	<*LibLocation*>	String	Project location
	<*PinDefInfo*>	Array	Optional, it defines pin location, type, and color. Multiple <*PinDefInfo*> parameters can be included. See below for more information.
	<*GraphicsDataInfo*>	Array	Optional; it defines graphics using one or more of the following: <RectInfo>, <CircleInfo>, <ArcInfo>, <LineInfo>, <PolygonInfo>, <TextInfo>, <ImageInfo>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			See below for more details.
	<PropDisplayMapInfo>	Array	Optional, it defines display properties. See below for more information.
	<ListOfComponentNames>	Array	The list may be empty. When not empty, each string that is listed is a component that references the symbol to be edited. Prior to editing, a clone of the component is made, and the components that are listed are modified so that they now refer to the clone.
Return Value	Composite name of the component in the form of a string. If the name requested conflicts with the name of an existing component, the requested name is altered to be unique, and the name returned reflects that change.		

Python Syntax	EditWithComps (*<ComponentName>*, [*<Name>*, *<ModTime>*, *<Library>*, *<LibLocation>*, *<PinDefInfo>*,*<GraphicsDataInfo>*, *<PropDisplayMapInfo>*],*<ListOfComponentNames>*)

Python Example	`oComponentManager.EditWithComps ("Nexxim Circuit Elements\Distributed\Distributed:bendo",`
	`  ["NAME:bendo new name",`
	`  "ModTime:=", 1152722165,`
	`  "Library:=", "Nexxim Circuit Elements\Distributed\Distributed",`
	`  "LibLocation:=", "SysLibrary",`
	`  ["NAME:PinDef",`
	`  "Pin:=", ["n1", -0.00254, 0.00254, 0, "N", 0, 0, false, 0, true, "",`
	`  false, false]`
	`  ],`
	`  ["NAME:PinDef",`
	`  "Pin:=", [ "n2", 0.00254, -0.00254, 1.5708, "N", 0, 0, false, 0, true, "",`
	`  false, false]`
	`  ],`
	`  ["NAME:Graphics",`
	`  "Polygon:=", [ 0, 0, 12566272, 0, 0.00381, 0, .00127, 0.00127, _ 0.00127,`
	`  0.00127, 0, 0.00381, 0, 0.00381, _ 0.002032, 0.00127, 0.00381],`
	`  "Line:=", [0, 1, 12566272, -0.00254, 0.00254, 0, .00254],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "Line:=", [0, 1, 12566272, 0.00254, -0.00254, .00254, 0]
    ],
    ["NAME:PropDisplayMap",
       "W:=", [3, 5, 0, "Text:=", [ 2.1684E-019, 0.00504119, 0, 1, 5, false,
      "Arial", 0, _ "W=***"]]
    ]
  ],
  ["MY_COMP"])
```

<PinDefInfo>:

```
["NAME:PinDef",
"Pin:=", [<string>, # pin name
<real>, # x location
<real>, # y location
<real>, # angle in radians
<PinType>,
<real>, # line width
<real>, # line length
<bool>, # mirrored
<int>, # color
<bool>, # True if visible, False if not
<string>, # hidden net name
<OptionalPinInfo>, # optional info
<PropDisplayMapInfo>]] # optional
```

<PinType> – defined as a string

```
# "N" : normal pin
# "I" : input pin
# "O" : output pin
```

<OptionalPinInfo>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# Specify both or neither
<bool>, # True if name is to be shown
<bool>, # True if number is to be shown
```

<PropDisplayMapInfo>: # optional

```
["NAME:PropDisplayMap", <PropDisplayInfo>, <PropDisplayInfo>,...]
```

<PropDisplayInfo>:

```
[ <NameString>, [<DisplayTypeInfo>,
<DisplayLocationInfo>,
<int>, # optional, level number
<TextInfo>]]
```

```
<NameString>: <string> # PropertyName:=, where PropertyName is the name of the property to
be displayed
```

<DisplayTypeInfo> – defined as an integer

```
# 0 : No display
# 1 : Display name only
# 2 : Display value only
# 3 : Display both name and value
# 4: Display evaluated value only
# 5: Display both name and evaluated value
```

<DisplayLocationInfo> – defined as an integer

```
# 0 : Left
# 1 : Top
# 2 : Right
# 3 : Bottom
# 4 : Center
# 5 : Custom placement
```

<GraphicsDataInfo>

```
["NAME:Graphics",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# one or more of the following
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>]
```

<RectInfo>:

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height
```

<CircleInfo>:

```
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius
```

<ArcInfo>:

```
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians
```

<LineInfo>:

```
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
<PointInfo>...]
```

<PointInfo>:

```
<real>, # x position
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```

<TextInfo>:

```
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string
```

<Justification> – defined as an integer

```
# 0 : left top
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    # 1 : left base
    # 2 : left bottom
    # 3 : center top
    # 4 : center base
    # 5 : center bottom
    # 6 : right top
    # 7 : right base
    # 8 : right bottom
```

<ImageInfo>:

```
"Image:=", [<RectInfo>,
<ImageData>,
<bool>] # is mirrored
```

<ImageData>:

```
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

Export [component manager]

Exports component(s) to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Components** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			the component to <*ComponentName*> – String that defines the composite name of the component to export
	<*LibraryLocation*>	String	Location to save the library. Only "PersonalLib" and "UserLib" are allowed.
Return Value	None.		

Python Syntax	Export(["NAME:<LibraryName>", <ComponentName>, <ComponentName>...], <LibraryLocation>)
Python Example	`oComponentManager.Export(["NAME:mylib",` `"Simplorer Circuit Elements\BJTs:Level01_NPN"],"PersonalLib")`

GetNames [component manager]

Returns the names of the components (used and unused) in a design. The following script command, **IsUsed,** can then be used to separate used and unused components.

UI Access	N/A
Parameters	None
Return Value	An array of strings

Python Syntax	GetNames()
Python Example	`componentNames = oComponentManager.GetNames()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNPortData [component manager]

Returns NPort data for the component with the specified name.

UI Access	Not available
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ComponentName*></td><td>String</td><td>The name of the component to be searched for NData.</td></tr></table>
Return Value	Variant array, whose contents depend on the type of component. The array will be empty if the component does not have NPort data. See the syntax for AddDynamicNPortData and AddNPortData for descriptions of the array contents for components with those types of NPort data.

Python Syntax	GetNPortData(<ComponentName>)
Python Example	`NPortdata = oComponentManager.GetNPortData("name")`

GetSolverOnDemandData

This method looks for a local component of the name passed in, and in that component, it looks for an SOD model of the name passed in and returns the SOD data pertaining to that model.

UI Access	Not available
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><BSTRName></td><td>String</td><td>BSTR component name</td></tr><tr><td><BSTRSODName></td><td>String</td><td>BSTR SOD model name</td></tr></table>
Return Value	VARIANT, which is the SOD data

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetSolverOnDemandData(<BSTRName>,<BSTRSODName>)
Python Example	

GetSolverOnDemandModelList

This method looks for a local component of the name passed in, and returns a list of SOD model names defined in the component.

UI Access	Not available
Parameters	Name | Type | Description <BSTRName> | String | BSTR component name
Return Value	VARIANT, which is a list of SOD model names

Python Syntax	GetSolverOnDemandModelList (<*BSTRName*>)
Python Example	

IsUsed [component manager]

Determines if a component is used in the design.

UI Access	N/A
Parameters	Name | Type | Description <*ComponentName*> | String | Component name to query
Return Value	Boolean; true if the specified component is used in the design

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	IsUsed(<*ComponentName*>)
Python Example	`IsUsed = oComponentManager.IsUsed("MyComponent")`

Remove [component manager]

Removes a component from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Components** > **Remove Component**
Parameters	See table below
Return Value	None

Name	Type	Description
<*ComponentName*>	String	Name of the component to remove
<*IsProjectComponent*>	Boolean	True if it is a project component
<*LibraryName*>	String	Name of the library
<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".

Python Syntax	Remove (<ComponentName>, <IsProjectComponent>, <LibraryName>, <LibraryLocation>)
Python Example	`oComponentManager.Remove ("Simplorer Circuit Elements\BJTs:Level01_NPN", _ true, "Project")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveSolverOnDemandModel

Use: This method looks for a local component of the name passed in, and in this component it looks for an SOD model of the name passed in and deletes the SOD model definition from the component.

Parameters: `BSTR component name.`

Parameters: `BSTR SOD model name`

Return Value: None.

RemoveUnused [component manager]

Removes components that are not used in the design.

UI Access	**Project** > **Remove Unused Definitions** is similar but operates slightly different and does not record script commands.
Parameters	None
Return Value	Boolean; True if one or more components are removed.

Note:

The order of calls to RemoveUnused is significant. As a result, removing definitions in an unordered fashion may cause other components in dependent definitions to be rendered unusable.

Also, the symbol and footprint of an unused component are not unusable until after the component itself is removed using the Component Manager Remove script.

Python Syntax	RemoveUnused()
Python Example	`removedComps = oComponentManager.RemoveUnused()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Update Dynamic Link [component manager]

Reads data from the linked design and updates the dynamic link component. This will update the following: properties, solutions, ports, geometry.

UI Access	**Dynamic Link Item RCM** > **Refresh Dynamic Link** or **Dynamic Link Component** in schematic **RCM** > **Refresh Dynamic Link** or click the **Refresh Dynamic Link** button in the **Footprint** tab of the Properties Window for selected Dynamic Link components in the Layout Editor.
Parameters	Name | Type | Description *<component name>* | String | Name of the dynamic link component
Return Value	None

Python Syntax	UpdateDynamicLink(<component name>)
Python Example	`oComponentManager.UpdateDynamicLink("TeeModel_L1")`

Footprint Manager Script Commands

The footprint manager provides access to footprints in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
oFootprintManager = oDefinitionManager.GetManager("Footprint")
```

The footprint manager script commands are listed below:

Add

Edit

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditWithComps

Export

GetNames

IsUsed

Remove

RemoveUnused

Add [footprint manager]

The general Add command tocan be used to add a footprint.

UI Access	**Tools** > **Edit Libraries** > **Footprints** > **Add Footprint**		
Parameters	Name	Type	Description
	<FootprintName>	String	Name of footprint to be created
	<ModTime>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
	<Library>	String	Library name
	<ModSinceLib>	Boolean	True if modified.
	<LibLocation>	String	Project location
	<OkayToMirror>	Boolean	True to allow mirroring
	<DependsOnSubstrateParams>	Boolean	True to make dependent on substrate parameters
	<NIdx>	Integer	Integer index
	<DefUnits>	String	Default length units to use if units are not specified in other parameters
	<LyrsArray>	Array	Includes *<LayerArray>* and *<StackupLayerArray>*; see below for more information on these parameters.
	<ActLyr>	String	Name of active layer
	<ToleranceArray>	Array	An array of three real numbers that specify distance tolerance,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			angle tolerance (in radians), and dimensionless tolerance.
	<PrimitivesInfo>		Optional; it defines primitives using one or more of the following: <RectInfo>, <CircleInfo>, <ArcInfo>, <LineInfo>, <PolygonInfo>, <TextInfo>, <ImageInfo> See below for more details.
	<PinsInfo>	Array	Optional; see below for more details
	<ViasInfo>	Array	Optional; see below for more details
	<EdgeportsInfo>,	Array	Optional; see below for more details
	<ComponentPropertyInfo>	Array	Optional; see below for more details
	<ScriptInfo>	Array	Optional; specified for scripted footprints; see details below.
Return Value	Returns a string of the composite name of the footprint. If the name requested conflicts with the name of an existing Footprint, the requested name is altered to be unique. The name returned reflects any change made to be unique.		

<LyrsArray>

<LayerArray>:

```
["N:=", <string>, # layer name
"ID:=", <int> ,
"T:=", <LayerTypeInfo>, # layer type
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"TB:=", <TopBottomInfo>,
"Col:=", <int>, # optional - color
"Pat:=", <int>, # optional - fill pattern
"Vis:=", <bool>, # optional - are objects on layer visible
"Sel:=", <bool>, # optional - are objects on layer selectable
"L:=", <bool>] # optional; are objects on layer locked (can't be edited)
```

<LayerTypeInfo> – defined as a string with one of the following: "signal", "dielectric", "metalized signal", "assembly", "silkscreen", "soldermask", "solderpaste", "glue", or "user".

<TopBottomInfo> – defined as a string with one of the following: "top", "neither", "bottom", or "template"

<StackupLayerArray>:

```
[<LayerArray>,

"Elev:=", <ElevationInfo>,

"SubL:=", ["Th:=", <Dimension>,# real number, may include units

"LElev:=", <Dimension>, # real number, may include units

"R:=", <Dimension>, # real number, may include units

"M:=", <MaterialInfo>, # name of layer material defined as a string]]
```

<ElevationInfo> – defined as a string with one of the following:

"top" - snap to top

"mid" - snap to middle

"bot" - snap to bottom

"edit" - manual edit

"none"

<PrimitivesInfo>

```
["NAME:Prims",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# one or more of the following
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>]
```

<RectInfo>:

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height
```

<CircleInfo>:

```
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius
```

<ArcInfo>:

```
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians
```

<LineInfo>:

```
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
<PointInfo>...]
```

<PointInfo>:

```
<real>, # x position
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```

<TextInfo>:

```
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string
```

<Justification> – defined as an integer

```
# 0 : left top
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    # 1 : left base
    # 2 : left bottom
    # 3 : center top
    # 4 : center base
    # 5 : center bottom
    # 6 : right top
    # 7 : right base
    # 8 : right bottom
```

<ImageInfo>:

```
"Image:=", [<RectInfo>,
<ImageData>,
<bool>] # is mirrored
```

<ImageData>:

```
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

<PinsInfo>:

```
["NAME:Pins",

"P:=", <PinArray>,

"P:=", <PinArray>,...]
```

<PinArray>:

```
["Port:=", ["Id:=", <int>,

"Clr:=", <real>, # optional - clearance

"N:=", <string>), # pin name
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Pos:=", ["x:=", <Location>, # padstack (x,y) position
"y:=", <Location>],
"VRt:=", <Angle>, # optional - rotation
"HD:=", <Size>, # optional - hole diameter
<PadstackImplementationInfo>]]
```

<Location> – string specifying real number and units (may use variables)

<Angle> – string specifying angle with a real number and units

<Size> – string specifying size with a real number and units

<PadstackImplementationInfo> – If another with the same implementation has already been specified:

```
"Ref:=", <int> # id of the other
```

If not:

```
"Ref:=", <string>, # name
"Frm:=", <int>, # id of highest layer
"To:=", <int>, # id of lowest layer
<LayerPlacementInfo>,
"Man:=", <int>, # optional, 1 if manually placed, 0 if not (default)
"Use:=", [<PadUseInfo>, <PadUseInfo>…] # array may be empty
```

<LayerPlacementInfo>:

```
"Lyr:=", ["Mrg:=", <int>, # optional,
# 1 if all layers have been merged (default)
# 0 if layer are not merged
"Flp:=", <int>, # optional,
# 1 if placed bottom up
# 0 if not (default)
"Map:=", <ParentToLocalLayerInfo>]
```

<ParentToLocalLayerInfo>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["U:=", <DirectionOfUniqueness>, # one of "forward", "backward", or "two ways"
"F:=", [<int>, <int>, …], # forward mapping; -1 is not mapped
"B:=", [<int>, <int>, …] # backward mapping; -1 is not mapped
]
```

<PadUseInfo>:

```
"Pad:=", ["Lid:=", <int>, # layer id
"T:=", <string>, # type : "connected", "thermal",
# "no_pad, "not_connected",
# or "not_connected_thermal"
"Man:=", <int> # optional; 1 if manually placed, 0 if not (default)
]
```

<**ViasInfo**>:

```
["NAME:Vias", <ViaInfo>, <ViaInfo>…]
```

<ViaInfo>:

```
"V:=", ["Id:=", <int>,
"N:=", <string>, # name
"Pos:=", ["x:=", <Location>, # via (x,y) position
"y:=", <Location>],
"VRt:=", <Angle>, # optional - rotation
<ImplementationInfo>]
```

<**EdgeportsInfo**>:

```
["NAME:EPorts", <EdgePortArray>, <EdgePortArray>…]
```

<EdgePortArray>

```
["NAME:EP",
"LP:=", ["Id:=", <int>, # port id
"N:=", <string>], # port name
"Eo:=", [<edge description>, <edge description>,...]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

]

<edge description> for primitive edges: "et:=", "pe", "pr:=", <id>, "ei:=", <edge#>

<id>: integer that is the primitive id

<edge#>: integer that is the edge number on the primitive

<edge description> for via edges:

"et:=", "pse", "sel:=", <"via">, "layer:=", <layer id>,

"sx:=", <start X location>, "sy:=", <start Y location>,

"ex:=", <end X location>, "ey:=", <end Y location>,

"h:=", <arc height>, "rad:=", <radians>

<"via">: text that is the name of the via to use

<layer id>: an integer that is the ID of the layer of the pad of the via to use

<start X location>, <start Y Location>: doubles that are the X, Y location of the start point of the edge arc

<end X location>, <end Y Location>: doubles that are the X, Y location of the end point of the edge arc

<arc height>: double giving the height of the edge arc (0 for a straight edge)

<radians>: double giving the arc size in radians (0 for a straight edge)

ComponentPropertyInfo>:

```
["NAME:CProps",
"VariableProp:=", <VariableInfo>,
"VariableProp:=", <VariableInfo>,
…]
```

<VariableInfo>:

```
[<string>, # name
<FlagLetters>,
<string>, # description
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "CB:=", <string>, # optional - script for call back
  <string>] # value: number, variable, or expression
```

<ScriptInfo>:

```
["NAME:script",
"language:=", <string>, # script language
"UsesScript:=", true,
"script:=", <string>] # contents of script
```

Python Syntax	Add([<FootprintName>, <ModTime>,<Library>,<ModSinceLib>, <LibLocation>, <OkayToMirror>, <DependsOnSubstrateParams>,<DefUnits>, <NIdx>, <Lyrs>,<ActLyr>)

Python Example	`oProject = oDesktop.SetActiveProject("SSN_ECAD_3DL")`
	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oFootprintManager = oDefinitionManager.GetManager("Footprint")`
	`oFootprintManager.Add(`
	`[`
	`  "NAME:Footprint2",`
	`  "ModTime:=" , 1746197177,`
	`  "Library:=" , "",`
	`  "ModSinceLib:=" , False,`
	`  "LibLocation:=" , "Project",`
	`  "OkayToMirror:=" , False,`
	`  "DependsOnSubstrateParams:=", False,`
	`  "DefUnits:=" , "mm",`
	`  "NIdx:=" , 0,`
	`  [`
	`  "NAME:Lyrs",`
	`  "Mode:=" , "Laminate",`
	`  [`
	`  "NAME:pps"`
	`  ],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"SLayer:=" , [ "Layer:=" , [ "N:=" , "Top", "ID:=" , 7, "T:=" , "signal",
"Col:=" , 32512, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127, "Zones:=" , []],
"SubL:=" , [ "Th:=" , "0mil", "LElev:=" , "62mil", "R:=" , "0um", "BR:=" ,
"0um", "SR:=" , "0um", "Mat:=" , "copper", "FilMat:=" , "FR4_epoxy"],
"UseR:=" , False, "RMdl:=" , "Huray", "NR:=" , "0mil", "HRatio:=" , "2.9",
"BRMdl:=" , "Huray", "BNR:=" , "0mil", "BHRatio:=" , "2.9", "SRMdl:=" ,
"Huray", "SNR:=" , "0mil", "SHRatio:=" , "2.9", "IsTSV:=" , False,
"OxideThicknesses:=" , [], "OxideMaterials:=" , []],
"SLayer:=" , [ "Layer:=" , [ "N:=" , "Dielectric", "ID:=" , 0, "T:=" ,
"dielectric", "Col:=" , 127, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127,
"Zones:=" , []], "SubL:=" , [ "Th:=" , "62mil", "LElev:=" , "0mil", "R:=" ,
"0um", "BR:=" , "0um", "SR:=" , "0um", "Mat:=" , "FR4"], "UseR:=" , False,
"RMdl:=" , "Huray", "NR:=" , "0mil", "HRatio:=" , "2.9", "BRMdl:=" ,
"Huray", "BNR:=" , "0mil", "BHRatio:=" , "2.9", "SRMdl:=" , "Huray", "SNR:="
 , "0mil", "SHRatio:=" , "2.9", "IsTSV:=" , False, "OxideThicknesses:=" ,
[], "OxideMaterials:=" , []],
"SLayer:=" , [ "Layer:=" , [ "N:=" , "Ground", "ID:=" , 6, "T:=" , "ground",
 "TB:=" , "bottom", "Col:=" , 4144959, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" ,
127, "Zones:=" , []], "SubL:=" , [ "Th:=" , "0mil", "LElev:=" , "0mil",
"R:=" , "0um", "BR:=" , "0um", "SR:=" , "0um", "Mat:=" , "copper",
"FilMat:=" , "FR4_epoxy"], "Neg:=" , True, "UseR:=" , False, "RMdl:=" ,
"Huray", "NR:=" , "0mil", "HRatio:=" , "2.9", "BRMdl:=" , "Huray", "BNR:=" ,
"0mil", "BHRatio:=" , "2.9", "SRMdl:=" , "Huray", "SNR:=" , "0mil", "SHRa-
tio:=" , "2.9", "IsTSV:=" , False, "OxideThicknesses:=" , [], "OxideMa-
terials:=" , []],
"Layer:=" , [ "N:=" , "SIwave Regions", "ID:=" , 8, "T:=" , "siwave-
hfsssolverregions", "Col:=" , 8355711, "Tr:=" , 60, "Pat:=" , 1, "Vf:=" ,
127],
"Layer:=" , [ "N:=" , "Rats", "ID:=" , 3, "T:=" , "rat", "Col:=" , 16711680,
 "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127],
"Layer:=" , [ "N:=" , "Errors", "ID:=" , 4, "T:=" , "error", "Col:=" , 255,
"Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127, "L:=" , True],
"Layer:=" , [ "N:=" , "Symbols", "ID:=" , 5, "T:=" , "symbol", "Col:=" ,
8323199, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127],
"Layer:=" , [ "N:=" , "Assembly Top", "ID:=" , 2, "T:=" , "assembly", "TB:="
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    , "top", "Col:=" , 16711680, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127,
    "Zones:=" , []],
    "Layer:=" , [ "N:=" , "Silkscreen Top", "ID:=" , 1, "T:=" , "silkscreen",
    "TB:=" , "top", "Col:=" , 65280, "Tr:=" , 0, "Pat:=" , 1, "Vf:=" , 127,
    "Zones:=" , []],
    "Layer:=" , [ "N:=" , "Postprocessing", "ID:=" , 9, "T:=" , "post-
    processing", "Col:=" , 9017384, "Tr:=" , 60, "Pat:=" , 1, "Vf:=" , 127],
    "Layer:=" , [ "N:=" , "Outline", "ID:=" , 10, "T:=" , "outline", "Col:=" ,
    8468789, "Tr:=" , 60, "Pat:=" , 1, "Vf:=" , 127],
    [
      "NAME:Locker",
      "IsLocked:=" , False
    ]
  ],
  "ActLyr:=" , "Top",
  [
      "NAME:Ncs",
      [
        "NAME:Nc",
        "Nam:=" , "<Power/Ground>",
        "Dsc:=" , ""
      ]
    ],
    [
      "NAME:Ncs",
      [
        "NAME:Nc",
        "Nam:=" , "<Power/Ground>",
        "Dsc:=" , ""
      ]
    ],
    [
      "NAME:Dps"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      ],
      [
         "NAME:VRMs"
      ]
])
```

Edit [footprint manager]

Deprecated command; please use EditWithComps.

EditWithComps [footprint manager]

Edit an existing footprint.

UI Access	Not available

Parameters

Name	Type	Description
<FootprintName>	String	Composite name of the model being edited
<FootprintPropertiesArray>	Array	Array of model properties
<NAME>	String	New simple name for the footprint
<ModTime>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
<Library>	String	Library name
<LibLocation>	String	Project location
<OkayToMirror>	Boolean	True to allow mirroring
<DefUnits>	String	Default length units to use if units are not specified in other parameters
<LyrArray>	Array	Defines layers and StackUp layers; see details below
<ActLyr>	String	Name of active layer
<ToleranceArray>	Array	Optional; An array of three real numbers that specify distance tolerance, angle tolerance (in radians), and dimensionless tolerance.
<PrimitivesInfo>		Optional; it defines primitives using one or more of the following: <RectInfo>, <CircleInfo>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<ArcInfo>, <LineInfo>, <PolygonInfo>, <TextInfo>, <ImageInfo> See below for more details.
	<PinsInfo>		Optional; see below for more details
	<ViasInfo>		Optional; see below for more details
	<EdgeportsInfo>		Optional; see below for more details
	<ComponentPropertyInfo>		
	<ScriptInfo>	Array	Optional; specified for scripted footprints; see details below.
	<ListOfComponentNames>	Array	The list may be empty. When not empty, each string that is listed is a component that references the footprint to be edited. Prior to editing, a clone of the footprint is made, and the components that are listed are modified so that they now refer to the clone.
Return Value	Composite name of the footprint in the form of a string. If the name requested conflicts with the name of an existing footprint, the requested name is altered to be unique, and the name returned reflects that change.		

<LyrsArray>

<LayerArray>:

```
["N:=", <string>, # layer name
"ID:=", <int> ,
"T:=", <LayerTypeInfo>, # layer type
"TB:=", <TopBottomInfo>,
"Col:=", <int>, # optional - color
"Pat:=", <int>, # optional - fill pattern
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Vis:=", <bool>, # optional - are objects on layer visible
"Sel:=", <bool>, # optional - are objects on layer selectable
"L:=", <bool>] # optional; are objects on layer locked (can't be edited)
```

<LayerTypeInfo> – defined as a string with one of the following: "signal", "dielectric", "metalized signal", "assembly", "silkscreen", "soldermask", "solderpaste", "glue", or "user".

<TopBottomInfo> – defined as a string with one of the following: "top", "neither", "bottom", or "template"

<StackupLayerArray>:

```
[<LayerArray>,

"Elev:=", <ElevationInfo>,

"SubL:=", ["Th:=", <Dimension>,# real number, may include units

"LElev:=", <Dimension>, # real number, may include units

"R:=", <Dimension>, # real number, may include units

"M:=", <MaterialInfo># name of layer material defined as a string]]
```

<ElevationInfo> – defined as a string with one of the following:

```
"top" - snap to top

"mid" - snap to middle

"bot" - snap to bottom

"edit" - manual edit

"none"
```

<PrimitivesInfo>

```
["NAME:Prims",
# one or more of the following
<RectInfo>,
<CircleInfo>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>]
```

<RectInfo>:

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height
```

<CircleInfo>:

```
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius
```

<ArcInfo>:

```
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<LineInfo>:

```
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
<PointInfo>...]
```

<PointInfo>:

```
<real>, # x position
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```

<TextInfo>:

```
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string
```

<Justification> – defined as an integer

```
# 0 : left top
# 1 : left base
# 2 : left bottom
# 3 : center top
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    # 4 : center base
    # 5 : center bottom
    # 6 : right top
    # 7 : right base
    # 8 : right bottom
```

<ImageInfo>:

```
"Image:=", [<RectInfo>,
<ImageData>,
<bool>] # is mirrored
```

<ImageData>:

```
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

<PinsInfo>:

```
["NAME:Pins",

"P:=", <PinArray>,

"P:=", <PinArray>,...]
```

<PinArray>:

```
["Port:=", ["Id:=", <int>,

"Clr:=", <real>, # optional - clearance

"N:=", <string>), # pin name

"Pos:=", ["x:=", <Location>, # padstack (x,y) position

"y:=", <Location>],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"VRt:=", <Angle>, # optional - rotation
"HD:=", <Size>, # optional - hole diameter
<PadstackImplementationInfo>]]
```

<Location> – string specifying real number and units (may use variables)

<Angle> – string specifying angle with a real number and units

<Size>– string specifying size with a real number and units

<PadstackImplementationInfo>:

If another with the same implementation has already been specified:

```
"Ref:=", <int> # id of the other
```

If not:

```
"Ref:=", <string>, # name
"Frm:=", <int>, # id of highest layer
"To:=", <int>, # id of lowest layer
<LayerPlacementInfo>,
"Man:=", <int>, # optional, 1 if manually placed, 0 if not (default)
"Use:=", [<PadUseInfo>, <PadUseInfo>…] # array may be empty
```

<LayerPlacementInfo>:

```
"Lyr:=", ["Mrg:=", <int>, # optional,
# 1 if all layers have been merged (default)
# 0 if layer are not merged
"Flp:=", <int>, # optional,
# 1 if placed bottom up
# 0 if not (default)
"Map:=", <ParentToLocalLayerInfo>]
```

<ParentToLocalLayerInfo>:

```
["U:=", <DirectionOfUniqueness>, # one of "forward", "backward", or "two ways"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"F:=", [<int>, <int>, …], # forward mapping; -1 is not mapped
"B:=", Array[<int>, <int>, …] # backward mapping; -1 is not mapped
]
```

<PadUseInfo>:

```
"Pad:=", ["Lid:=", <int>, # layer id
"T:=", <string>, # type : "connected", "thermal",
# "no_pad, "not_connected",
# or "not_connected_thermal"
"Man:=", <int> # optional; 1 if manually placed, 0 if not (default)
]
```

<ViasInfo>:

```
["NAME:Vias", <ViaInfo>, <ViaInfo>…]
```

<ViaInfo>:

```
"V:=", ["Id:=", <int>,
"N:=", <string>, # name
"Pos:=", ["x:=", <Location>, # via (x,y) position
"y:=", <Location>],
"VRt:=", <Angle>, # optional - rotation
<ImplementationInfo>]
```

<EdgeportsInfo>:

```
["NAME:EPorts", <EdgePortArray>, <EdgePortArray>…]
```

<EdgePortArray>

```
["NAME:EP",
"LP:=", ["Id:=", <int>, # port id
"N:=", <string>], # port name
"Eo:=", [<edge description>, <edge description>,...]
]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<edge description> for primitive edges: "et:=", "pe", "pr:=", <id>, "ei:=", <edge#>

<id>: integer that is the primitive id

<edge#>: integer that is the edge number on the primitive

<edge description> for via edges:

"et:=", "pse", "sel:=", <"via">, "layer:=", <layer id>,

"sx:=", <start X location>, "sy:=", <start Y location>,

"ex:=", <end X location>, "ey:=", <end Y location>,

"h:=", <arc height>, "rad:=", <radians>

<"via">: text that is the name of the via to use

<layer id>: an integer that is the ID of the layer of the pad of the via to use

<start X location>, <start Y Location>: doubles that are the X, Y location of the start point of the edge arc

<end X location>, <end Y Location>: doubles that are the X, Y location of the end point of the edge arc

<arc height>: double giving the height of the edge arc (0 for a straight edge)

<radians>: double giving the arc size in radians (0 for a straight edge)

<ComponentPropertyInfo>:

```
["NAME:CProps",
"VariableProp:=", <VariableInfo>,
"VariableProp:=", <VariableInfo>,
…]
```

<VariableInfo>:

```
[<string>, # name
<FlagLetters>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  <string>, # description
  "CB:=", <string>, # optional - script for call back
  <string>] # value: number, variable, or expression
```

<ScriptInfo>:

```
["NAME:script",
"language:=", <string>, # script language
"UsesScript:=", true,
"script:=", <string>] # contents of script
```

Python Syntax	EditWithComps (<FootprintName>, [<*Name*>, <*ModTime*>, <*Library*>, <*LibLocation*>, <*OkayToMirror*>,<*DefUnits*>, <*LyrsArray*>, <*ActLyr*>, <*ToleranceArray*>, <*PrimitivesInfo*>,<*PinsInfo*>, <*ViasInfo*>,<*EdgeportsInfo*>,<*ComponentPropertyInfo*>, <*ScriptInfo*>],<*ListOfComponentNames*>)

Python Example	`oFootprintManager.EditWithComps \("Nexxim Circuit Elements\Distributed\Distributed:bendo",`
	`  [NAME:bendo new name",`
	`  "ModTime:=", 1152722165,`
	`  "Library:=", "Nexxim Circuit Elements\Distributed\Distributed",`
	`  "LibLocation:=", "SysLibrary",`
	`  ["NAME:PinDef",`
	`  "Pin:=", [ "n1", -0.00254, 0.00254, 0, "N", 0, 0, false, 0, true, "", false, false]`
	`  ],`
	`  ["NAME:PinDef",`
	`  "Pin:=", [ "n2", 0.00254, -0.00254, 1.5708, "N", 0, 0, false, 0, true, "", false, false]`
	`  ],`
	`  ["NAME:Graphics",`
	`  "Polygon:=", [ 0, 0, 12566272, 0, 0.00381, 0, .00127, 0.00127, 0.00127,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      0.00127, 0, 0.00381, 0, 0.00381, 0.002032, 0.00127, 0.00381],
      "Line:=", [0, 1, 12566272, -0.00254, 0.00254, 0, .00254],
      "Line:=", [0, 1, 12566272, 0.00254, -0.00254, .00254, 0]
    ],
    ["NAME:PropDisplayMap",
      "W:=", [3, 5, 0,
        "Text:=", [ 2.1684E-019, 0.00504119, 0, 1, 5, false, "Arial", 0,
        "W=***"]
      ]
    ]
  ],
  ["MY_COMP"])
```

Export [footprint manager]

Export a footprint to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Footprints** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the footprint to <*FootprintName*> – String that defines the composite name of the footprint to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are "Project", "PersonalLib", or "UserLib".
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Export(*["NAME:<LibraryName>"*, <*FootprintName*>*]*, <*LibraryLocation*>)
Python Example	`oFootprintManager.Export(["NAME:mylib", "Distributed Footprints:BPAD"], "PersonalLib")`

GetNames [footprint manager]

Returns the names of the footprints (used and unused) in a design. The **IsUsed** script command can then be used to separate used and unused footprints.

UI Access	N/A
Parameters	None
Return Value	An array of strings

Python Syntax	GetNames()

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oFootprintManager = oDefinitionManager.GetManager("Footprint")`
	`footprintNames = oFootprintManager.GetNames()`

IsUsed [footprint manager]

Used to determine if a footprint is used in the design.

UI Access	N/A
Parameters	

Name	Type	Description
<*FootprintName*>	String	Footprint name to query

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Boolean; true if the specified footprint is used in the design

Python Syntax	IsUsed(<*FootprintName*>)

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oFootprintManager = oDefinitionManager.GetManager("Footprint")`
	`IsUsed = ooFootprintManager.IsUsed("MyFootprint")`

Remove [footprint manager]

Removes a footprint from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Footprints** > **Remove Footprint**
Parameters	See table below
Return Value	None

Name	Type	Description
<*FootprintName*>	String	Composite name of the footprint to remove
<*IsProjectFootprint*>	Boolean	True if it is a project footprint
<*LibraryName*>	String	Name of the library
<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".

Python Syntax	Remove (<*FootprintName*>, <*IsProjectFootprint*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oFootprintManager.Remove("BPAD", false, "MyLib", "PersonalLib")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveUnused [footprint manager]

Removes footprints that are not used in the design.

UI Access	**Project** > **Remove Unused Definitions** is similar but operates slightly different and does not record script commands.
Parameters	None
Return Value	Boolean; True if one or more footprints are removed.

> **Note:**
>
> The order of calls to RemoveUnused is significant. As a result, removing definitions in an unordered fashion may cause other footprints in dependent definitions to be rendered unusable.
>
> Also, the symbol and footprint of an unused component are not unusable until after the component itself is removed using the Component ManagerRemove script.

Python Syntax	RemoveUnused()

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oFootprintManager = oDefinitionManager.GetManager("Footprint")`
	`removed = oFootprintManager.RemoveUnused()`

Material Manager Script Commands

The material manager provides access to materials in a project. The manager object is accessed via the definition manager.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oDefinitionManager = oProject.GetDefinitionManager()
oMaterialManager = oDefinitionManager.GetManager("Material")
```

The topics for this section include:

GetNames

GetProperties

IsUsed

RemoveUnused

GetNames [material manager]

Gets the names of the materials in a project.

UI Access	N/A
Parameters	None
Return Value	Names of the materials in a project

Python Syntax	GetNames()
Python Example	`materialnames = oMaterialManager.GetNames()`

GetProperties [material manager]

Get material properties. Differs from GetData in that only material properties available to the user are returned by GetProperties.

UI Access	NA
Parameters	Name: <material_name> — Type: string — Description: Name of the project material

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		Boolean	True or False. If you use False or only a single argument, then GetProperties returns only the properties that are different from the defaults.
Return Value	Array of material data		

Python Syntax	GetProperties (<materialname>)
Python Example	`oMaterialManager.GetProperties("Gold", True)` `oMaterialManager.GetProperties("vacuum")`

IsUsed [material manager]

Checks if a project material is in use.

UI Access	N/A		
Parameters	Name	Type	Description
	<*MaterialName*>	String	Name of the project material to check; should include "library name:material name".
Return Value	Boolean; True if the specified material is used in the design.		

Python Syntax	IsUsed(<ComboName>)
Python Example	`isused = oMaterialManager.IsUsed("mylib:mymaterial")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveUnused [material manager]

Removes all unused materials from the project.

UI Access	None
Parameters	None
Return Value	None

Python Syntax	RemoveUnused()
Python Example	`thereWereExtras = oMaterialManager.RemoveUnused()`

Model Manager Script Commands

The model manager provides access to models in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
oModelManager = oDefinitionManager.GetManager("Model")
```

The model manager script commands are listed below:

Add

ConvertToDynamic

ConvertToParametric

Edit

EditWithComps

Export

GetNames

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IsUsed

Remove

RemoveUnused

Add [model manager]

Adds a model.

<table>
<tr><td>UI Access</td><td>Tools > Edit Configured Libraries > Models > Add Model</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><modelName></td><td>String</td><td>Name of the model being added</td></tr>
<tr><td><ModTime></td><td>Integer</td><td>An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.</td></tr>
<tr><td><Library></td><td>String</td><td>Library name</td></tr>
<tr><td><LibLocation></td><td>String</td><td>Name of project and file path</td></tr>
<tr><td><PinDefInfo></td><td>Array</td><td>Optional, defines a model pins; multiple <PinDefInfo> arrays can be included; contents of the <PinDefInfo> array are detailed below.</td></tr>
<tr><td><GraphicsDataInfo></td><td>Array</td><td>Optional, defines graphics. The array contains the following:
["NAME:Graphics",
one or more of the following
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<TextInfo>, <ImageInfo>] The individual array attributes are detailed below.
	<PropDisplayMapInfo>	Array	Optional, defines property displays; contents of the <PropDisplayMapInfo> are detailed below.
Return Value	Returns a string that includes composite name of the model. If the name requested conflicts with the name of an existing model, the requested name is altered to be unique, and the name returned reflects that alteration.		

Details of the `<PinDefInfo>` array:

```
["NAME:PinDef",
  "Pin:=", [<string>, # pin name
  <real>, # x location
  <real>, # y location
  <real>, # angle in radians
  <PinType>,
  <real>, # line width
  <real>, # line length
  <bool>, # mirrored
  <int>, # color
  <bool>, # True if visible, False if not
  <string>, # hidden net name
  <OptionalPinInfo>, # optional info
  <PropDisplayMapInfo>]] # optional

  <PinType>:
  <string> # "N" : normal pin
  # "I" : input pin
  # "O" : output pin

  <OptionalPinInfo>:
  # Specify both or neither
  <bool>, # True if name is to be shown
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<bool>, # True if number is to be shown

<PropDisplayMapInfo>:
["NAME:PropDisplayMap",
<PropDisplayInfo>,
<PropDisplayInfo>,...]

<PropDisplayInfo>:
<NameString>, [<DisplayTypeInfo>,
<DisplayLocationInfo>,
<int>, // optional, level number
<TextInfo>]
<NameString>:
<string> # PropertyName:=, where PropertyName is the name of the property to be displayed

<DisplayTypeInfo>:
<int> # 0 : No display
# 1 : Display name only
# 2 : Display value only
# 3 : Display both name and value
# 4: Display evaluated value only
# 5: Display both name and evaluated value

<DisplayLocationInfo>:
<int> # 0 : Left
# 1 : Top
# 2 : Right
# 3 : Bottom
# 4 : Center
# 5 : Custom placement
```

Details for arrays used in `<GraphicsDataInfo>`:

```
<RectInfo>:
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height

<CircleInfo>:
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius

 <ArcInfo>:
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians

<LineInfo>:
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
<PointInfo>...)
<PointInfo>:
<real>, # x position
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<real>] # y position

<PolygonInfo>:
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]

<TextInfo>:
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string

<Justification>:
<int> # 0 : left top
# 1 : left base
# 2 : left bottom
# 3 : center top
# 4 : center base
# 5 : center bottom
# 6 : right top
# 7 : right base
# 8 : right bottom

<ImageInfo>:
"Image:=", [<RectInfo>,
<ImageData>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<bool>] # is mirrored

<ImageData>:
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

Python Syntax	Add (["NAME:<modelName>",<ModTime>, "Library", "LibLocation",<PinDefInfo>,<GraphicsDataInfo>, <PropDisplayMapInfo>])
Python Example	oModelManager.Add(["NAME:MyModel", "ModTime:=", 1070989137, "Library:=", "", "LibLocation:=", "Project", ["NAME:PinDef", "Pin:=", ["newpin0", 0.00254, 0, 0, "N",0, 0.00254, false, 0, true, "" ,false, false]], ["NAME:PinDef", "Pin:=", ["newpin1", -0.00254, 0, 3.14159265358979," N", 0, 0.00254, false, 0, true, "", false, false]], ["NAME:Graphics", "Rect:=", [0, 0, 12566272, 0, 4.33680868994202e-019, - 0.000635,0.00508, 0.002794], "Circle:=", [0, 0, 12566272, 0.000127, -0.000635, 0.000635]]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ConvertToDynamic

Use: Build a new dynamic model based on an existing parametric model.

Command: Right-click on a model under Definitions/Models in the Project Tree and choose ConvertToDynamic.

Syntax: ConvertToDynamic(defName, newname)

Return Value: <newname> // Name of the new model added

Parameters: <defName> // Model that is the base for the new conversion

ConvertToParametric

Use: Build a new parametric model based on an existing dynamic model.

Command: Right-click on a model under Definitions/Models in the Project Tree and choose ConvertToParametric.

Syntax: ConvertToParametric(defName, newname)

Return Value: <newname> // Name of the new model added

Parameters: <defName> // Model that is the base for the new conversion

Edit [deprecated]

Deprecated command; please use EditSymbolAndUpdateComps.

EditWithComps [model manager]

Edit an existing model.

UI Access	Not available		
Parameters	Name	Type	Description
	<ModelName>	String	Composite name of the model being edited
	<ModelPropertiesArray>	Array	Array of model properties

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<NAME>	String	New simple name for the model
	<ModTime>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
	<Library>	String	Library name
	<LibLocation>	String	Project location
	<PinDefInfo>	Array	Optional, it defines pin location, type, and color. Multiple *<PinDefInfo>* parameters can be included. See below for more information.
	<GraphicsDataInfo>	Array	Optional; it defines graphics using one or more of the following: <RectInfo>, <CircleInfo>, <ArcInfo>, <LineInfo>, <PolygonInfo>, <TextInfo>, <ImageInfo> See below for more details.
	<PropDisplayMapInfo>	Array	Optional, it defines display properties. See below for more information.
	<ListOfComponentNames>	Array	The list may be empty. When not empty, each string that is listed is a component that references the model to be edited. Prior to editing, a clone of the model is made, and the components that are listed are modified so that they now refer to the clone.
Return Value	Composite name of the model in the form of a string. If the name requested conflicts with the name of an existing model, the requested name is altered to be unique, and the name returned reflects that change.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditWithComps (<*ModelName*>, [<*Name*>, <*ModTime*>, <*Library*>, <*LibLocation*>, <*PinDefInfo*>,<*GraphicsDataInfo*>, <*PropDisplayMapInfo*>],<*ListOfComponentNames*>)
Python Example	oModelManager.EditWithComps ("Nexxim Circuit Elements\Distributed\Distributed:bendo", ["NAME:bendo new name", "ModTime:=", 1152722165, "Library:=", "Nexxim Circuit Elements\Distributed\Distributed", "LibLocation:=", "SysLibrary", ["NAME:PinDef", "Pin:=", ["n1", -0.00254, 0.00254, 0, "N", 0, 0, false, 0, true, "", false, false]], ["NAME:PinDef", "Pin:=", ["n2", 0.00254, -0.00254, 1.5708, "N", 0, 0, false, 0, true, "", false, false]], ["NAME:Graphics", "Polygon:=", [0, 0, 12566272, 0, 0.00381, 0, .00127, 0.00127, _ 0.00127, 0.00127, 0, 0.00381, 0, 0.00381, _ 0.002032, 0.00127, 0.00381], "Line:=", [0, 1, 12566272, -0.00254, 0.00254, 0, .00254], "Line:=", [0, 1, 12566272, 0.00254, -0.00254, .00254, 0]], ["NAME:PropDisplayMap", "W:=", [3, 5, 0, "Text:=", [2.1684E-019, 0.00504119, 0, 1, 5, false, "Arial", 0, _ "W=***"]]]],

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["MY_COMP"])

<PinDefInfo>:

```
["NAME:PinDef",
"Pin:=", [<string>, # pin name
<real>, # x location
<real>, # y location
<real>, # angle in radians
<PinType>,
<real>, # line width
<real>, # line length
<bool>, # mirrored
<int>, # color
<bool>, # True if visible, False if not
<string>, # hidden net name
<OptionalPinInfo>, # optional info
<PropDisplayMapInfo>]] # optional
```

<PinType> – defined as a string

```
# "N" : normal pin
# "I" : input pin
# "O" : output pin
```

<OptionalPinInfo>:

```
# Specify both or neither
<bool>, # true if name is to be shown
<bool>, # true if number is to be shown
```

<PropDisplayMapInfo>: # optional

```
["NAME:PropDisplayMap", <PropDisplayInfo>, <PropDisplayInfo>,...]
```

<PropDisplayInfo>:

```
[ <NameString>, [<DisplayTypeInfo>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<DisplayLocationInfo>,
<int>, # optional, level number
<TextInfo>]]
```

```
<NameString>: <string> # PropertyName:=, where PropertyName is the name of the property to
be displayed
```

<DisplayTypeInfo> – defined as an integer

```
# 0 : No display
# 1 : Display name only
# 2 : Display value only
# 3 : Display both name and value
# 4: Display evaluated value only
# 5: Display both name and evaluated value
```

<DisplayLocationInfo> – defined as an integer

```
# 0 : Left
# 1 : Top
# 2 : Right
# 3 : Bottom
# 4 : Center
# 5 : Custom placement
```

<GraphicsDataInfo>

```
["NAME:Graphics",
# one or more of the following
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<RectInfo>:

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height
```

<CircleInfo>:

```
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius
```

<ArcInfo>:

```
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians
```

<LineInfo>:

```
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<PointInfo>...]
```

<PointInfo>:

```
<real>, # x position
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```

<TextInfo>:

```
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string
```

<Justification> – defined as an integer

```
# 0 : left top
# 1 : left base
# 2 : left bottom
# 3 : center top
# 4 : center base
# 5 : center bottom
# 6 : right top
# 7 : right base
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  # 8 : right bottom
```

<ImageInfo>:

```
"Image:=", [<RectInfo>,
<ImageData>,
<bool>] # is mirrored
```

<ImageData>:

```
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

Export [model manager]

Exports model(s) to a library.

<table>
<tr><td>UI Access</td><td colspan="3">Tools > Edit Configured Libraries > Models > Export to Library</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><*NameArray*></td><td>Array</td><td>Includes the following:

<*LibraryName*> – String that defines the name of the library to export the model to

<*ModelName*> – String that defines the composite name of the model to export; multiple models can be included.</td></tr>
<tr><td><*LibraryLocation*></td><td>String</td><td>Location of the library specified with <LibraryName>; options are "Project", "PersonalLib", or "UserLib".</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Export (["*NAME:<LibraryName>*", <*ModelName*>, <*ModelName*>...], <*LibraryLocation*>)
Python Example	`oModelManager.Export([NAME:Nexxim Circuit Elements\Distributed\Distributed:bendo:mylib", "myModel"], "PersonalLib)`

GetNames [model manager]

Returns the names of the models (used and unused) in a design. The following script command, **IsUsed,** can then be used to separate used and unused models.

UI Access	N/A
Parameters	None
Return Value	An array of strings

Python Syntax	GetNames()

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oModelManager = oDefinitionManager.GetManager("Model")`
	`modelnames = oModelManager.GetNames()`

IsUsed [model manager]

Used to determine if a model is used in the design.

UI Access	N/A
Parameters	Name | Type | Description <*ModelName*> | String | Component name to query, in the form of LibraryName:ModelName

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Boolean; true if the specified model is used in the design

Python Syntax	IsUsed(<*ModelName*>)
Python Example	`isused = oModelManager.IsUsed ("mylib:mymodel")`

Remove [model manager]

Removes a model from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Models** > **Remove Model** button
Parameters	

Name	Type	Description
<*ModelName*>	String	Specifies the composite name of the model to remove
<*IsProjectModel*>	Boolean	True if it is a project model
<*LibraryName*>	String	Name of the library if it is not a project model
<*LibraryLocation*>	String	Location of the library in <LibraryName>; one of "Project", "PersonalLib", or "UserLib"

Return Value	None

Python Syntax	Remove (<ModelName>, <IsLocal>, <LibraryName>, <LibraryLocation>)
Python Example	`oSimModelManager = oDefinitionManager.GetManager("SimModel")` `oSimModelManager.Remove("simmodel", True, "", "Project")`

RemoveUnused [model manager]

Removes models that are not used in the design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Project** > **Remove Unused Definitions** is similar but operates slightly different and does not record script commands.
Parameters	None
Return Value	Boolean; True if one or more modes are removed.

> **Note:**
>
> The order of calls to RemoveUnused is significant. As a result, removing definitions in an unordered fashion may cause other models in dependent definitions to be rendered unusable.
>
> Also, the model and footprint of an unused component are not unusable until after the component itself is removed using the Component Manager Remove script.

Python Syntax	RemoveUnused()
Python Example	`thereWereExtras = oModelManager.RemoveUnused()`

Network Data Explorer Manager Script Commands

The network data Explorer (NDE) Manager provides access to certain NDE data.

```
oDefinitionManager = oProject.GetDefinitionManager()
oNdExplorerManager = oDefinitionManager.GetManager("NdExplorer")
```

For NDE scripts accessed via the ndExplorer tool, see Network Data Explorer Script Commands.

The topics for this section include

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportFullWaveSpice

ExportNetworkData

ExportNMFData

ExportFullWaveSpice

Export FullWaveSpice data in a format of your choice.

<table>
<tr><td>UI Access</td><td>File > Export MacroModel > Broadband (SYZ, FWS….)</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td>DesignName</td><td>String</td><td>Design name. Can be left blank if loading solution from a file.</td></tr>
<tr><td>true/false</td><td>Bool</td><td>true - solution loaded from file; false - loaded from design</td></tr>
<tr><td>Name</td><td>String</td><td>If loading from design, this is the solution name; else, full path of the file</td></tr>
<tr><td>variation</td><td></td><td>Pick a particular variation. Leave blank if none.</td></tr>
<tr><td>["NAME:Frequencies"]</td><td>Array</td><td>Optional; if none defined, all frequencies are used</td></tr>
<tr><td>["NAME:SpiceData"]</td><td>Array</td><td>Spice export options object. The following parameters are for this array.
<table>
<tr><td>SpiceType</td><td>"SSS" - Options: "PSpice", "HSpice", "Spectre", "SSS", "Simplorer", "TouchStone1.0", "TouchStone2.0"</td></tr>
<tr><td>EnforcePassivity</td><td>false - Enforce passivity (true/false)</td></tr>
<tr><td>EnforceCausality</td><td>false - Enforce causality (true/false)</td></tr>
<tr><td>UseCommonGround</td><td>false - Use common ground (true/false)</td></tr>
<tr><td>FittingError</td><td>0.5 - Fitting error</td></tr>
<tr><td>MaxPoles</td><td>400 - Maximum order</td></tr>
<tr><td>PassivityType</td><td>"ConvexOptimization" - Options: "ConvexOptimization", "PassivityByPerturbation", "IteratedFittingOfPV", "IteratedFittingOfPVLF"</td></tr>
<tr><td>ColumnFittingType</td><td>"Column" - Options: "Column", "Entry", "Mat-</td></tr>
</table></td></tr>
</table></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

				rix"
			SSFittingType	"TWA" - Options: "TWA", "IterativeRational"
			RelativeErrorToleranc	false - Relative error tolerance (true/false)
			TouchstoneFormat	"MA" - Options: "MA", "RI", "DB"
			TouchstoneUnits	"Hz" - Options: "Hz", "KHz", "MHz"
			TouchStonePrecision	8 - Touchstone precision
			ExportDirectory	"C:/Examples/LNA/" - Directory to export to
			ExportSpiceFileName	"Linckt_HBTest_2.sss" - Spice export file name
			FullwaveSpiceFileName	"Linckt_HBTest.sss" - Fullwave Spice file name
			CreateNPortModel	true - Create a model based on the exported file (true/false)
Return Value	[none]			

Python Syntax	ExportFullWaveSpice(<DesignName> , <True/False> , <Name> , <Variation> , <[NAME:Frequencies]> , <[NAME:SpiceData]>)

Python Example	`oTool.ExportFullWaveSpice("", False, "", "$isolation=\'0.13\' p1=\'8.7062500000000007mm\' p2=\'8.6100700000000003mm\' s1=\'0.41463699999999998mm\' s2=\'1.40411mm\' w1=\'0.44363399999999997mm\' w2=\'0.47960799999999998mm\' w50=\'1.0031099999999999mm\'",`
	`  [`
	`  "NAME:Frequencies",`
	`  4500000000,`
	`  4505000000,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    4510000000,
    4515000000,
    4520000000,
    4525000000,
    4530000000,
    4535000000,
    4540000000,
    4545000000
  ],
  [
    "NAME:SpiceData",
    "SpiceType:=" , "HSpice",
    "EnforcePassivity:=" , False,
    "EnforceCausality:=" , False,
    "UseCommonGround:=" , True,
    "ShowGammaComments:=" , False,
    "Renormalize:=" , False,
    "RenormImpedance:=" , 50,
    "FittingError:=" , 0.5,
    "MaxPoles:=" , 10000,
    "PassivityType:=" , "IteratedFittingOfPVLF",
    "ColumnFittingType:=" , "Matrix",
    "SSFittingType:=" , "FastFit",
    "RelativeErrorToleranc:=", False,
    "EnsureAccurateZfit:=" , True,
    "TouchstoneFormat:=" , "MA",
    "TouchstoneUnits:=" , "GHz",
    "TouchStonePrecision:=" , 15,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "SubcircuitName:=" , "",
        "ExportDirectory:=" , "C:/Program Files/ANSYS Inc/v261/An-
        sysEM/Examples/Circuit/RF Microwave/Filters/",
        "ExportSpiceFileName:=" , "filter_layout_LinearFrequency__isolation_0_13_
        p1_8_70625mm_p2_8_61007mm_s1_0_414637mm_s2_1_40411mm_w1_0_443634mm_w2_0_
        479608mm_w50_1_00311mm_1_sp.sp",
        "FullwaveSpiceFileName:=", "filter_layout_LinearFrequency__isolation_0_
        13_p1_8_70625mm_p2_8_61007mm_s1_0_414637mm_s2_1_40411mm_w1_0_443634mm_w2_
        0_479608mm_w50_1_00311mm_1_sp.sp",
        "UseMultipleCores:=" , True,
        "NumberOfCores:=" , 10
])
```

ExportNetworkData

Exports matrix solution data to a file.

UI Access	N/A		
Parameters	Name	Type	Description
	<DesignVariationKey>	String	Design variation key. Pass empty string for the current nominal variation.
	<SolnSelectionArray>	Array	Array of selected solutions. `[<SolnSelector>,<SolnSelector>, ...]` If more than one array entry, this indicates a combined Interpolating sweep.
	<SolnSelector>	String	Solution setup name and solution name, separated by a colon.
	<FileFormat>	Integer	File format value. 2 : Tab delimited spreadsheet format (.tab)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		3 : Touchstone (.sNp) 4 : CitiFile (.cit) 7 : Matlab (.m) 8 : Terminal Z0 spreadsheet
<*OutFile*>	String	Full path to the file to write out.
<*FreqsArray*>	Array	The frequencies to export. The <FreqsArray> argument contains a vector (e.g. "1GHz", "2GHz", ...) to use, or "all". To export all frequencies, use ["all"]. If no frequencies are specified, all frequencies are used.
<*DoRenorm*>	Boolean	For Touchstone format only. Specifies whether to renormalize the data before export.
<*RenormImped*>	Double	For Touchstone format only. Real impedance value in ohms, for renormalization. Required in syntax, but ignored if DoRenorm is false.
<*DataType*>	Array	Optional. Type: "S", "Y", or "Z". The matrix to export.
<*pass*>	Integer	Optional. The pass to export. This is ignored if the sourceName is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
<*ComplexFormat*>	Integer	Optional. Type: "0", "1", or "2" The format to use for the exported data. 0 = Magnitude/Phase. 1= Real/Imaginary. 2= db/Phase.
<*Precision*>	Integer	Optional. Touchstone number of digits precision. Default if not specified is 15.
<*UseExportFreqs*>	Boolean	Specifies whether to use export frequencies.
<*IncludeGammaComments*>	Boolean	Touchstone only. Specifies whether to include Gamma and Impedance comments.
<*SupportNonStdExport*>	Boolean	Specifies whether to support non-standard Touchstone extensions

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	for mixed reference impedance.
Return Value	None.

Python Syntax	ExportNetworkData(<*DesignVariationKey*>, <*SolnSelectionArray*>, <*SolnSelector*>, <*FileFormat*>, <*OutFile*>, <*FreqsArray*>, <*DoRenorm*>, <*RenormImped*>, <*DataType*>, <*pass*>, <*ComplexFormat*>, <*Precision*>, <*UseExportFreqs*>, <*IncludeGammaComments*>, <*SupportNonStdExport*>)
Python Example	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

ExportNMFData

Exports s-parameters in neutral file format (*.nmf).

UI Access	**Results** > **Solution Data**. On the **Matrix** tab, click **Export** tab. Then click **Export** > **S-parameter**. Select Neutral Model format and export.
Parameters	

Name	Type	Description
<*sourceName*>	Array	String names of solutions. Use more than one array entry for a combined Interpolating sweep.
<*outputFileName*>	String	Full path of output file, including *.nmf extension.
<*reduceMat*>	String	Either "Original" or name of reduced matrix.
<*refImp*>	Double	Reference impedence.
<*freqs*>	Array	Either "All" or array of doubles representing frequencies.
<*variationKey*>	String	Design variation key. Pass empty string for the current nominal variation.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<displayFormat>	String	"MagPhase", "RealImag", "DbPhase".
	<pass>	Integer	Optional. The pass to export. This is ignored if the *sourceName* is a frequency sweep. Leaving out this value or specifying -1 gets all passes.
	<ComplexFormat>	Integer	Optional. The format to use for the exported data. Options are 0 = Magnitude/Phase 1= Real/Imaginary. 2= db/Phase.
	<Precision>	Integer	Optional. Number of digits precision. If not specified, the default is 6.
Return Value	None.		

Python Syntax	ExportNMFData (*<sourceName>*, *<outputFileName>*,*<reduceMat>*, *<reflmp>*, *<freqs>*,*<variationKey>*, *<displayFormat>*, *<pass>*, *<ComplexFormat>, <Precision>*)
Python Example	`oDesign.ExportNMFData("Setup2 : LastAdaptive", "C:/outputs/data.nmf, "Original", 50, [10000000, 20000000, 30000000, 40000000, 50000000, 60000000, 70000000, 80000000, 90000000, 100000000], "", "MagPhase", 0, 1, 11)`

Padstack Manager Script Commands

The padstack manager provides access to padstacks in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
```

```
oPadstackManager = oDefinitionManager.GetManager("Padstack")
```

The padstack manager script commands are listed below.

Add

Edit

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditWithComps

Export

GetNames

IsUsed

Remove

RemoveUnused

Add [padstack manager]

Adds a padstack.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Add Padstack**		
Parameters	Name	Type	Description
	<NAME>	String	Simple name of padstack to create; in the form of "Name:PadstackName"
	<ModifiedOn>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
	<Library>	String	Padstack library.
	<LibLocation>	String	Location of the padstack.
	<psdArray>	Array	Includes "Name:psd" and the subparameters listed below.
	<nam>	String	<PadstackName>
	<lib>	String	Name of the library
	<mat>	String	Hole plating material
	<plt>	String	Percent of hole's radius filled by plating
	<pdsArray>	Array	Includes "Name:pds" and a list of *<LayerGeometryArray>* that defines the padstack geometry. See below for more details this array.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<hle>	Array	Defines <PadInfo>; see below for more details on this array.
	<hRg>	String	Defines the hole range. One of the following choices: "Thr" – through all layout layers "Beg" – from upper pad layer to lowest layout layer "End" – from upper layout layer to lowest pad layer "UTL" – from upper pad layer to lowest pad layer
	<sbsh>	String	Defines the solder ball shape. One of the following choices: "None" – no solderball "Cyl" – cylinder solderball "Sph" – spheroid solderball
	<sbpl>	String	Defines the solder ball placement. One of the following choices: "abv" – above padstack "blw" – below padstack
	<sbr>	String	Solderball diameter; real number with units
	<sb2>	String	Solderball mid diameter, real number with units
	<sbn>	String	Name of solderball material
	<PadPortLayerArray>	Array	List of integers where each integer is a layer id
Return Value	Simple name of the added padstack. If the name requested conflicts with the name of an existing padstack, the requested name is altered to be unique, and the name returned reflects any change made.		

<LayerGeometryArray>:

```
["Name:lgm",
"lay:=", <string>, // definition layer name
"id:=", <int>, // definition layer id
"pad:=", <PadInfo>, // pad
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"ant:=", <PadInfo>, // antipad
"thm:=", <PadInfo>, // themal pad
"X:=", <string>, // pad x connection, real with units
"Y:=", <string>, // pad y connection, real with units
"dir:=", <DirectionString>] // pad connection direction
```

<PadInfo>:

```
["shp:=", <PadShape>,
"Szs:=", <DimensionArray>,
"X:=", <string>, // x offset, real with units
"Y:=", <string>, // y offset, real with units
"R:=", <string>] // rotation, real with units
```

<PadShape>:

One of these choices:

```
"No" // no pad
"Cir" // Circle
"Sq" // Square
"Rct" // Rectangle
"Ov" // Oval
"Blt" // Bullet
"Ply" // Polygons
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"R45" // Round 45 thermal
"R90" // Round 90 thermal
"S45" // Square 45 thermal
"S90" // Square 90 thermal
```

<DimensionArray>:

An array of strings; each string is a real with units for one of the dimensions of the shape

<DirectionString>:

One of these choices:

```
"No" // no direction
"Any" // any direction
"0" // 0 degrees
"45" // 45 degrees
"90" // 90 degrees
"135" // 135 degrees
"180" // 180 degrees
"225" // 225 degrees
"270" // 270 degrees
"315" // 315 degrees
```

Python Syntax	oPadstackManager.Add(<*PadstackName*>, <*ModifiedOnInfo*>, <*psdArray*>,<PadPortLayerArray>)
Python Example	`oPadstackManager.Add(` `  ["NAME:Circle - through3",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"ModTime:=", 1235765635,
"Library:=", "",
"LibLocation:=", "Project",
["NAME:psd", "nam:=", "Circle - through3", "lib:=", "", "mat:=", "",
"plt:=", "0",
  ["NAME:pds",
    ["NAME:lgm", "lay:=", "Top Signal", "id:=", 0,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2.5mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", ["shp:=", "Cir", "Szs:=", ["3.5mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0"],
      "thm:=", ["shp:=", "No", "Szs:=", [], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0"],
       "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
    ["NAME:lgm", "lay:=", "SignalA", "id:=", 1,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", ["shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0"],
      "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0"],
      "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
    ["NAME:lgm", "lay:=", "SignalB", "id:=", 2,
      "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "ant:=", [shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=",
      "0mm", "R:=", "0deg"],
      "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "R:=", "0"],
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
      ["NAME:lgm", "lay:=", "Ground", "id:=", 3,
        "pad:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0deg"],
        "ant:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0"],
        "thm:=", ["shp:=", "R90", "Szs:=", ["3mm", "0.75mm", "1mm"], "X:=",
        "0mm", "Y:=", "0mm", "R:=", "0"],
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
      ["NAME:lgm", "lay:=", "Bottom signal", "id:=", 5,
        "pad:=", ["shp:=", "Cir", "Szs:=", ["1mm"], "X:=", "0mm", "Y:=",
        "0mm", "R:=", "0deg"],
        "ant:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=",
        "0mm", "R:=", "0deg"],
        "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
        "R:=", "0"],
        "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"]
    ]
    "hle:=", ["shp:=", "Cir", "Szs:=", ["1.5mm"], "X:=", "0mm", "Y:=", "0mm",
    "R:=", "0deg"],
    "hRg:=", "End",
    "sbsh:=", "Sph",
    "sbpl:=", "abv",
    "sbr:=", "750um",
    "sb2:=", "1200um", "1200um",
    "sbn:=", "solder"],
  "ppl:=", [0, 1, 2, 3, 5]
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit [padstack manager]

Use: Edit an existing padstack.

Command: Tools > Edit Configured Libraries > Padstacks > Edit Padstack

Syntax: Edit <PadstackName>,

```
Array("NAME:<NewPadstackName>",
"ModTime:=", <ModifiedOnInfo>,
"Library:=", "", // name of the library
"LibLocation:=", "Project", // location of the named library
Array("NAME:psd",
"nam:= ", <PadstackName>,
"lib:=", "", // name of the library
"mat:=", "", // hole plating material
"plt:=", "0", // percent of hole's radius filled by plating
Array("NAME:pds",
<LayerGeometryArray>,
<LayerGeometryArray....),
"hle:=", <PadInfo>
"hRg:=", <HoleRange>,
"sbsh:=", <SolderballShape>,
"sbpl:=", <SolderballPlacement>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"sbr:=", <string>, // solderball diameter, real with units

"sb2:=", <string>, // solderball mid diameter, real with units

"sbn:=", <string>), // name of solderball material

"ppl:=", <PadPortLayerArray>)

Return Value: <string> // composite name of the padstack

// If the name requested conflicts with the name of an existing

// padstack, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <PadstackName>:

<string> // composite name of padstack to edit

<NewPadstackName>:

<string> // new simple name for padstack

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed

since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<LayerGeometryArray>:

Array("Name:lgm",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential

information of ANSYS, Inc. and its subsidiaries and affiliates.

"lay:=", <string>, // definition layer name

"id:=", <int>, // definition layer id

"pad:=", <PadInfo>, // pad

"ant:=", <PadInfo>, // antipad

"thm:=", <PadInfo>, // themal pad

"X:=", <string>, // pad x connection, real with units

"Y:=", <string>, // pad y connection, real with units

"dir:=", <DirectionString>) // pad connection direction

<PadInfo>:

Array("shp:=", <PadShape>,

"Szs:=", <DimensionArray>,

"X:=", <string>, // x offset, real with units

"Y:=", <string>, // y offset, real with units

"R:=", <string>) // rotation, real with units

<PadShape>:

<string> one of these choices

"No" // no pad

"Cir" // Circle

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Sq" // Square

"Rct" // Rectangle

"Ov" // Oval

"Blt" // Bullet

"Ply" // Polygons

"R45" // Round 45 thermal

"R90" // Round 90 thermal

"S45" // Square 45 thermal

"S90" // Square 90 thermal

<DimensionArray>:

Array(<string>, ...) // each string is a real with units for one of the

// dimensions of the shape

<DirectionString>:

<string> one of these choices

"No" // no direction

"Any" // any direction

"0" // 0 degrees

"45" // 45 degrees

"90" // 90 degrees

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"135" // 135 degrees

"180" // 180 degrees

"225" // 225 degrees

"270" // 270 degrees

"315" // 315 degrees

<HoleRange>:

<string> one of these choices

"Thr" // through all layout layers

"Beg" // from upper pad layer to lowest layout layer

"End" // from upper layout layer to lowest pad layer

"UTL" // from upper pad layer to lowest pad layer

<SolderballShape>:

<string> one of these choices

"None" // no solderball

"Cyl" // cylinder solderball

"Sph" // spheroid solderball

<SolderballPlacement>:

<string> one of these choices

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"abv" // above padstack

"blw" // below padstack

<PadPortLayerArray>:

Array(<int>, <int>,....) where each int is a layer id

EditWithComps [padstack manager]

Edit an existing padstack.

<table>
<tr><td>UI Access</td><td>Not available</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PadstackName></td><td>String</td><td>Composite name of the Padstack being edited</td></tr>
<tr><td><PadstackPropertiesArray></td><td>Array</td><td>Array of Padstack properties</td></tr>
<tr><td><NAME></td><td>String</td><td>New simple name for the Padstack</td></tr>
<tr><td><ModTime></td><td>Integer</td><td>An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.</td></tr>
<tr><td><Library></td><td>String</td><td>Library name</td></tr>
<tr><td><LibLocation></td><td>String</td><td>Project location</td></tr>
<tr><td><psdArray></td><td>Array</td><td>["NAME:psd",
"nam:= ", <PadstackName>,
"lib:=", "", #name of the library
"mat:=", "", # hole plating material
"plt:=", "0", # percent of hole's radius filled by plating]</td></tr>
<tr><td><lgmArray></td><td>Array</td><td>Layer information in the form of
["NAME:lgm",
<LayerGeometryArray>,
<LayerGeometryArray....</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			], See the details below.
	<PadInfo>	Array	Optional, it defines display properties. See below for more information.
	<HoleRange>	String	One of these choices: "Thr" – through all layout layers "Beg" – from upper pad layer to lowest layout layer "End" – from upper layout layer to lowest pad layer "UTL" – from upper pad layer to lowest pad layer
	<SolderballShape>	String	One of these choices: "None" – no solderball "Cyl" – cylinder solderball "Sph" – spheroid solderball
	<SolderballPlacement>	String	One of these choices: "abv" – above padstack "blw" – below padstack
	<sbr>	String	Solderball diameter, real with units
	<sb2>	String	Solderball mid diameter, real with units
	<sbn>	String	Name of solderball material
	<PadPortLayerArray>	Array	List of integers where each integer is a layer ID.
	<ListOfComponentNames>	Array	The list may be empty. When not empty, each string that is listed is a component that references the padstack to be edited. Prior to editing, a clone of the padstack is made, and the components that are listed are modified so that they now refer to the clone.
Return Value	Composite name of the padstack in the form of a string. If the name requested conflicts with the name of an exist-		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	ing padstack , the requested name is altered to be unique, and the name returned reflects that change.

Python Syntax	EditWithComps (<*PadstackName*>, [<*Name*>, <*ModTime*>, <*Library*>, <*LibLocation*>, <*PinDefInfo*>,<*GraphicsDataInfo*>, <*PadPortLayerArray*>],<*ListOfComponentNames*>)

Python Example	`oPadstackManager.EditWithComps("Circle - through1",`
	`  ["NAME:Circle - through1",`
	`  "ModTime:=", 1235765635,`
	`  "Library:=", "",`
	`  "LibLocation:=", "Project",`
	`  ["NAME:psd", "nam:=", "Circle - through1", "lib:=", "", "mat:=", "", "plt:=",`
	`  "0",`
	`  ["NAME:pds",`
	`  ["NAME:lgm", "lay:=", "Top Signal", "id:=", 0,`
	`  "pad:=", ["shp:=", "Cir", "Szs:=", ["2.5mm"], "X:=", "0mm", "Y:=",`
	`  "0mm", "R:=", "0deg"],`
	`  "ant:=", ["shp:=", "Cir", "Szs:=", ["3.5mm"], "X:=", "0mm", "Y:=",`
	`  "0mm", "R:=", "0"],`
	`  "thm:=", ["shp:=", "No", "Szs:=", [], "X:=", "0mm", "Y:=", "0mm", "R:=",`
	`  "0"],`
	`  "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],`
	`  ["NAME:lgm", "lay:=", "SignalA", "id:=", 1,`
	`  "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=", "0mm",`
	`  "R:=", "0deg"],`
	`  "ant:=", ["shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=", "0mm",`
	`  "R:=", "0"],`
	`  "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",`
	`  "R:=", "0"],`
	`  "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ["NAME:lgm", "lay:=", "SignalB", "id:=", 2,
          "pad:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0deg"],
          "ant:=", [shp:=", "Cir", "Szs:=", ["3mm"], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0deg"],
          "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0"],
          "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
        ["NAME:lgm", "lay:=", "Ground", "id:=", 3,
          "pad:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0deg"],
          "ant:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0"],
          "thm:=", ["shp:=", "R90", "Szs:=", ["3mm", "0.75mm", "1mm"], "X:=",
          "0mm", "Y:=", "0mm", "R:=", "0"],
          "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"],
        ["NAME:lgm", "lay:=", "Bottom signal", "id:=", 5,
          "pad:=", ["shp:=", "Cir", "Szs:=", ["1mm"], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0deg"],
          "ant:=", ["shp:=", "Cir", "Szs:=", ["2mm"], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0deg"],
          "thm:=", ["shp:=", "No", "Szs:=", [ ], "X:=", "0mm", "Y:=", "0mm",
          "R:=", "0"],
          "X:=", "0mm", "Y:=", "0mm", "dir:=", "Any"]
      ]
      "hle:=", ["shp:=", "Cir", "Szs:=", ["1.5mm"], "X:=", "0mm", "Y:=", "0mm",
      "R:=", "0deg"],
      "hRg:=", "End",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"sbsh:=", "Sph", "sbpl:=", "abv", "sbr:=", "750um", "sb2:=", "1200um", "1200um", "sbn:=", "solder"], "ppl:=", [0, 1, 2, 3, 5] [""])

<LayerGeometryArray>:

```
["Name:lgm",
"lay:=", <string>, // definition layer name
"id:=", <int>, // definition layer id
"pad:=", <PadInfo>, // pad
"ant:=", <PadInfo>, // antipad
"thm:=", <PadInfo>, // themal pad
"X:=", <string>, // pad x connection, real with units
"Y:=", <string>, // pad y connection, real with units
"dir:=", <DirectionString>] // pad connection direction
```

<PadInfo>:

```
["shp:=", <PadShape>,
"Szs:=", <DimensionArray>,
"X:=", <string>, // x offset, real with units
"Y:=", <string>, // y offset, real with units
"R:=", <string> // rotation, real with units
]
```

<PadShape>– Input is a string; use one of the following:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"No" // no pad
"Cir" # Circle
"Sq" # Square
"Rct" # Rectangle
"Ov" # Oval
"Blt" # Bullet
"Ply" # Polygons
"R45" # Round 45 thermal
"R90" # Round 90 thermal
"S45" # Square 45 thermal
"S90" # Square 90 thermal
```

<DimensionArray>:

```
[<string>, ...] # each string is a real with units for one of the dimensions of the shape
```

<DirectionString>– Input is a string; use one of the following:

```
"No" # no direction
"Any" # any direction
"0" # 0 degrees
"45" # 45 degrees
"90" # 90 degrees
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"135" # 135 degrees
"180" # 180 degrees
"225" # 225 degrees
"270" # 270 degrees
"315" # 315 degrees
```

Export [padstack manager]

Exports a padstack to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the padstack to <*PadstackName*> – String that defines the simple name of the padstack to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are `"Project"`, `"PersonalLib"`, or `"UserLib"`.
Return Value	None		

Python Syntax	Export(*[Name:*<*LibraryName*>, <*PadstackName*>], <*LibraryLocation*>)
Python Example	`oPadstackManager.Export(["NAME:mylib", "myPadstack"], "PersonalLib")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNames [padstack manager]

Returns the names of the padstack (used and unused) in a design. The **IsUsed** script command can then be used to separate used and unused padstacks.

UI Access	N/A
Parameters	None
Return Value	An array of strings

Python Syntax	GetNames()

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oPadstackManager = oDefinitionManager.GetManager("Padstack")`
	`padstackNames = oPadstackManager.GetNames()`

IsUsed [padstack manager]

Used to determine if a padstack is used in the design.

UI Access	N/A
Parameters	Name | Type | Description <*PadstackName*> | String | Padstack name to query
Return Value	Boolean; true if the specified padstack is used in the design

Python Syntax	IsUsed(<*PadstackName*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oPadstackManager = oDefinitionManager.GetManager("Padstack")`
	`IsUsed = oPadstackManager.IsUsed("MyPadstack")`

Remove [padstack manager]

Removes a padstack from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Padstacks** > **Remove Padstacks**
Parameters	See table below
Return Value	None

Name	Type	Description
<*PadstackName*>	String	Simple name of the padstack to remove
<*IsProjectPadstack*>	Boolean	True if it is a project padstack
<*LibraryName*>	String	Name of the library
<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".

Python Syntax	Remove (<PadstackName>, <IsProjectPadstack>, <LibraryName>, <LibraryLocation>)
Python Example	`oPadstackManager.Remove ("Polygon SMT", false, "MyLib", "PersonalLib")`

RemoveUnused [padstack manager]

Removes padstacks that are not used in the design.

UI Access	**Project** > **Remove Unused Definitions** is similar but operates slightly different and does not record script commands.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None
Return Value	Boolean; True if one or more padstacks are removed.

Note:

The order of calls to RemoveUnused is significant. As a result, removing definitions in an unordered fashion may cause other padstacks in dependent definitions to be rendered unusable.

Python Syntax	RemoveUnused()
Python Example	`removedStacks = oPadstackManager.RemoveUnused()`

Script and Library Scripts

The definition manager provides access to materials in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
```

The script and library script commands are listed below.

AddScript

EditScript

ExportScript

RemoveScript

ModifyLibraries

AddScript

Adds a script to the definition manager.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A

Parameters	Name: *<AddScriptArray>* — Type: Array — Description: Structured array.
	`["NAME:<string script name>",`
	`"ScriptLang:=", <string script language>`
	`"ScriptText:=, <string text of script>]`

Return Value	None.

Python Syntax	AddScript(*<AddScriptArray>*)

Python Example	`oDefinitionManager.AddScript(`
	`["NAME:MyScript",`
	`"ScriptLang:=", "language",`
	`"ScriptText:=", "MsgBox(\"HelloWorld\")"`
	`]`
	`)`

EditScript

Edits a script in the definition manager.

UI Access	N/A
Parameters	Name — Type — Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*OriginalName*>	String	Name of the script to be edited.

	<*EditScriptArray*>	Array	Structured array.
			`["NAME:<string script name>",`
			`"ScriptLang:=", <string script language>`
			`"ScriptText:=, <string text of script>]`

Return Value	None.		

Python Syntax	EditScript(<*OriginalName*>, <*EditScriptArray*>)

Python Example	`oDefinitionManager.EditScript("MyScript",`
	`["NAME:MyNewScript",`
	`"ScriptLang:=", "language",`
	`"ScriptText:=", "MsgBox(\"HelloAgain\")"`
	`]`
	`)`

ExportScript

Exports a script to a library in the script definition manager.

UI Access	**None**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	Name	Type	Description
	<*ScriptArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the script to <*ScriptName*> – String that defines the name of the script to export
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are "`Project`", "`PersonalLib`", or "`UserLib`".
Return Value	None		

Python Syntax	ExportScript(<*ScriptArray*>,<*LibraryLocation*>)
Python Example	`oProject.ExportScript(["NAME:mylib", "myscript"], "PersonalLib")`

ModifyLibraries

Configures libraries accessed from the **Tools** menu.

UI Access	N/A		
Parameters	Name	Type	Description
	<*DesignName*>	String	Name of specified design.

	<*ConfigLibArray*>	Array	`["NAME:<LibraryType>,<ConfiguredLib>,<ConfiguredLib>,…]`
			`<ConfiguredLib> # blank to leave unchanged`
			`<DefinitionType>`
			`["<libraryname >","<libraryname>",…]`

Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ModifyLibraries (<*DesignName*>,<*ConfigLibArray*>)

Python Example	`oDefinitionManager.ModifyLibraries(`
	`  "MyCircuit",`
	`  ["NAME:PersonalLib"],`
	`  ["NAME:UserLib"],`
	`  ["NAME:SystemLib",`
	`  "Symbols:=", [ "Circuit Elements", "Symbols",`
	`  "ParamExtraElements\PE_Symbols",`
	`  "Vendor Elements\Nonlinear"]`
	`  ])`

RemoveScript

Removes a script in the script definition manager.

UI Access	**None**		
Parameters	Name	Type	Description
	<*ScriptName*>	String	Simple name of the script to remove
	<*IsProjectScript*>	Boolean	True if it is a project script
	<*LibraryName*>	String	Name of the library
	<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".
Return Value	None		

Python Syntax	RemoveScript (<*ScriptName*>,<*IsProjectScript*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveScript(`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"myscript", true, "Local", "Project")

Symbol Manager Script Commands

The symbol manager provides access to symbols in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
oSymbolManager = oDefinitionManager.GetManager("Symbol")
```

The symbol manager script commands are listed below.

Add

BringToFront

Edit

EditSymbolAndUpdateComps

Export

GetNames

IsUsed

Remove

RemoveUnused

Add [symbol manager]

Use: Add a symbol

Command: Tools > Edit Configured Libraries > Symbols > Add Symbol

Syntax: Add Array("NAME:<SymbolName>",

"ModTime:=", <ModifiedTimeInfo>,

"Library:=", "", // Library name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"LibLocation:=", "Project", // Project Location

<PinDefInfo>,

<PinDefInfo>,... // optional, to define pins

<GraphicsDataInfo>, // optional, to define graphics

<PropDisplayMapInfo>)) // optional, to define property displays

Return Value: <string>

// composite name of the symbol.

// If the name requested conflicts with the name of an existing

// symbol, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <SymbolName>:

<string> // simple name of the symbol being added

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed

since 00:00 hours, Jan 1, 1970 UTC from the system clock.

<PinDefInfo>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Array("NAME:PinDef",
"Pin:=", Array (<string>, // pin name
<real>, // x location
<real>, // y location
<real>, // angle in radians
<PinType>,
<real>, // line width
<real>, // line length
<bool>, // mirrored
<int>, // color
<bool>, // true if visible, false if not
<string>, // hidden net name
<OptionalPinInfo>, // optional info
<PropDisplayMapInfo>)) // optional

<PinType>:
<string> // "N" : normal pin
// "I" : input pin
// "O" : output pin

<OptionalPinInfo>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

// Specify both or neither
<bool>, // true if name is to be shown
<bool>, // true if number is to be shown

<PropDisplayMapInfo>:
Array("NAME:PropDisplayMap",
<PropDisplayInfo>,
<PropDisplayInfo>,...)

<PropDisplayInfo>:
<NameString>, Array(<DisplayTypeInfo>,
<DisplayLocationInfo>,
<int>, // optional, level number
<TextInfo>)
<NameString>:
<string> // PropertyName:=, where PropertyName is the name of
// the property to be displayed

<DisplayTypeInfo>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<int> // 0 : No display

// 1 : Display name only

// 2 : Display value only

// 3 : Display both name and value

// 4: Display evaluated value only

// 5: Display both name and evaluated value

<DisplayLocationInfo>:

<int> // 0 : Left

// 1 : Top

// 2 : Right

// 3 : Bottom

// 4 : Center

// 5 : Custom placement

<GraphicsDataInfo>:

Array("NAME:Graphics",

// one or more of the following

<RectInfo>,

<CircleInfo>,

<ArcInfo>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<LineInfo>,

<PolygonInfo>,

<TextInfo>,

<ImageInfo>)

<RectInfo>:

"Rect:=", Array(<real>, // line width

<int>, // fill pattern

<int>, // color

<real>, // angle, in radians

<real>, // x position of center

<real>, // y position of center

<real>, // width

< real>) // height

<CircleInfo>:

"Circle:=", Array(<real>, // line width

<int>, // fill pattern

<int>, // color

<real>, // x position of center

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<real>, // y position of center

< real>) // radius

<ArcInfo>:

"Arc:=", Array(<real>, // line width

<int>, // line pattern

<int>, // color

<real>, // x position of center

<real>, // y position of center

< real>, // radius

<real>, // start angle, in radians

<end>) // end angle, in radians

<LineInfo>:

"Line:=", Array(<real>, // line width

<int>, // line pattern

<int>, // color

<PointInfo>, // must specify at least 2 points

<PointInfo>...)

<PointInfo>:

<real>, // x position

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<real> // y position

<PolygonInfo>:
"Polygon:=", Array(<real>, // line width
<int>, // fill pattern
<int>, // color
<PointInfo>, // must specify at least 3 points
<PointInfo>...)

<TextInfo>:
"Text:=", Array(<real>, // x position
<real>, // y position
<real>, // angle, in radians
<Justification>,
<bool>, // is plotter font
<string>, // font name
<int>, // color
<string>) // text string

<Justification>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<int> // 0 : left top

// 1 : left base

// 2 : left bottom

// 3 : center top

// 4 : center base

// 5 : center bottom

// 6 : right top

// 7 : right base

// 8 : right bottom

<ImageInfo>:

"Image:=", Array(<RectInfo>,

<ImageData>,

<bool>) // is mirrored

<ImageData>:

<string>, // file path

<int>, // 0 : use the file path and link to it

// 1 : ignore file path and use next parameter

<string> // text data, only present if preceding int is 1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

BringToFront [symbol manager]

Changes the drawing for the symbol so that the specified objects are drawn on top of other overlapping objects.

UI Access	**Draw** > **Bring to Front**		
Parameters	Name	Type	Description
	<Name>	String	"NAME:Selections"
	<*Selections*>	Array	List of objects (strings) to bring to the front
Return Value	None		

Python Syntax	BringToFront([<*Name*>, <*Selections*>])

Python Example	`oDefinitionEditor.BringToFront(`
	`  ["NAME:Selections",`
	`  "Selections:=", ["SchObj@10"]`
	`  ])`

Edit [deprecated]

Deprecated command; please use EditSymbolAndUpdateComps.

EditSymbolAndUpdateComps [symbol manager]

Edits an existing symbol; it includes a <*RefPinOption*> parameter, which can be used to define changes made ref pins.

UI Access	Not available

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*SymbolName*>	String	Name of symbol to be edited
<*SymbolPropertiesArray*>	Array	Array of symbol properties
<*NAME*>	String	New simple name for the symbol
<*ModTime*>	Integer	An integer that corresponds to the number of seconds that have elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.
<*Library*>	String	Library name
<*LibLocation*>	String	Project location
<*PinDefInfo*>	Array	Optional, it defines pin location, type, and color. Multiple <*PinDefInfo*> parameters can be included. See below for more information.
<*GraphicsDataInfo*>	Array	Optional; it defines graphics using one or more of the following: <RectInfo>, <CircleInfo>, <ArcInfo>, <LineInfo>, <PolygonInfo>, <TextInfo>, <ImageInfo> See below for more details.
<*PropDisplayMapInfo*>	Array	Optional, it defines display properties. See below for more information.
<*ListOfComponentNames*>	Array	The list may be empty. When not empty, each string that is listed is a component that references the symbol to be edited. Prior to editing, a clone of the symbol is made, and the components that are listed are modified so that they now refer to the clone.
<*EditContext*>	Array	Optional, it defines the changes that will be made to the component in support of the symbol changes. It includes <*RefPinOption*>, <*CompName*>, and <*TermAttributes*>; see the details below.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	The composite name of the symbol. If the name requested conflicts with the name of an existing symbol, the requested name is altered to be unique, and the name returned reflects those changes.

Python Syntax	EditSymbolAndUpdateComps (<*SymbolName*>, [<*Name*>, <*ModTime*>, <*Library*>, <*LibLocation*>, <*PinDefInfo*>,<*GraphicsDataInfo*>, <*PropDisplayMapInfo*>],<*ListOfComponentNames*>, <*EditContext*>)
Python Example	

```
oDefinitionManager = oProject.GetDefinitionManager()
oSymbolManager = oDefinitionManager.GetManager("Symbol")
oSymbolManager.EditSymbolAndUpdateComps("Amp_Full",
  [
    "NAME:Amp_Full",
    "ModTime:=" , 1729524400,
    "CE:=" , 0,
    "Library:=" , "",
    "ModSinceLib:=" , False,
    "LibLocation:=" , "Project",
    "HighestLevel:=" , 1,
    "Normalize:=" , False,
    "InitialLevels:=" , [0,1],
    [
      "NAME:PinDef",
      "Pin:=" , ["Port1",-
      0.00508,0,0,"N",0,0.00254,False,0,True,"",True,False,"Port1",True],
      [
        "NAME:PropDisplayMap",
        "PinName:=" , [2,5,1, "Text:=" , [-0.00508,0.000635,0,4,5,False,"Ari-
        al",0,"Port1",False,False, "ExtentRect:=" , [0,0,0,0,-
        0.00508,0.00116761958303845,0.00379225143123376,0.00176829701568765,0,0,0]]]
      ]

  ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:PinDef",
  "Pin:=" , ["Port2",
  0.00508,0.001,3.14159265358979,"N",0,0.00254,False,0,True,"",True,False,"Port2",Tr-
  ue],
  [
    "NAME:PropDisplayMap",
    "PinName:=" , [2,5,1, "Text:=" , [0.00508,-0.005,0,4,5,False,"Ari-
    al",0,"Port2",False,False, "ExtentRect:=" ,
    [0,0,0,0,0.00508,0.00370761958303845,0.00379225143123376,0.00176829701568765,0,-
    0,0]]]
  ]
],
[
  "NAME:Graphics",
  "Rect:=" , [0,0,0,0,0,0,0.00508,0.00508,0,0,0],
  "Rect:=" , [0,1,0,0,-
  0.00211666666666667,0,0.000423333333333333,0.000423333333333334,0,0,0]
],
[
  "NAME:PropDisplayMap",
  "FileName:=" , [4,3,0, "Text:=" , [0,0,0,4,5,False,"Ari-
  al",0,"FileName",False,False, "ExtentRect:=" ,
  [0,0,0,0,0,0.000532619583038449,0.00679622587957061,0.00176829701568765,0,0,0]]]
]

], [],

["NAME:EditContext", "RefPinOption:=", 0, "CompName:=", "Model",

  ["NAME:Terminals",

    "TermAttributes:=", [ "M_DQ_0__AL11_IPD031-201_U2A5", "M_DQ_0__AL11_IPD031-201_
    U2A5", 0, 0, -1, ""],

    "TermAttributes:=", [ "M_DQ_0__AL11_IPD031-201_U2A5_ref", "M_DQ_0__AL11_IPD031-
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        201_U2A5_ref", 1, 0, -1, ""],
        "TermAttributes:=", ["M_DQ_0__B3_G83568-001_U1B5", "M_DQ_0__B3_G83568-001_U1B5",
        1, 0, -1, ""],
        "TermAttributes:=", [ "M_DQ_0__B3_G83568-001_U1B5_ref", "M_DQ_0__B3_G83568-001_
        U1B5_ref", 1, 0, -1, ""]
      ]
    ])
oDefinitionEditor.CloseEditor()
```

<PinDefInfo>:

```
["NAME:PinDef",
"Pin:=", [<string>, # pin name
<real>, # x location
<real>, # y location
<real>, # angle in radians
<PinType>,
<real>, # line width
<real>, # line length
<bool>, # mirrored
<int>, # color
<bool>, # true if visible, false if not
<string>, # hidden net name
<OptionalPinInfo>, # optional info
<PropDisplayMapInfo>]] # optional
```

<PinType> – defined as a string

```
# "N" : normal pin
# "I" : input pin
# "O" : output pin
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<OptionalPinInfo>:

```
# Specify both or neither
<bool>, # True if name is to be shown
<bool>, # True if number is to be shown
```

<PropDisplayMapInfo>:

```
["NAME:PropDisplayMap", <PropDisplayInfo>, <PropDisplayInfo>,...]
```

<PropDisplayInfo>:

```
[ <NameString>, [<DisplayTypeInfo>,
<DisplayLocationInfo>,
<int>, # optional, level number
<TextInfo>]]
```

```
<NameString>: <string> # PropertyName:=, where PropertyName is the name of the property to
be displayed
```

<DisplayTypeInfo> – defined as an integer

```
# 0 : No display
# 1 : Display name only
# 2 : Display value only
# 3 : Display both name and value
# 4: Display evaluated value only
# 5: Display both name and evaluated value
```

<DisplayLocationInfo> – defined as an integer

```
# 0 : Left
# 1 : Top
# 2 : Right
# 3 : Bottom
# 4 : Center
# 5 : Custom placement
```

<GraphicsDataInfo>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:Graphics",
# one or more of the following
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>]
```

<RectInfo>:

```
"Rect:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # angle, in radians
<real>, # x position of center
<real>, # y position of center
<real>, # width
<real>] # height
```

<CircleInfo>:

```
"Circle:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<real>, # x position of center
<real>, # y position of center
<real>] # radius
```

<ArcInfo>:

```
"Arc:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<real>, # x position of center
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<real>, # y position of center
<real>, # radius
<real>, # start angle, in radians
<end>] # end angle, in radians
```

<LineInfo>:

```
"Line:=", [<real>, # line width
<int>, # line pattern
<int>, # color
<PointInfo>, # must specify at least 2 points
<PointInfo>...]
```

<PointInfo>:

```
<real>, # x position
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```

<TextInfo>:

```
"Text:=", [<real>, # x position
<real>, # y position
<real>, # angle, in radians
<Justification>,
<bool>, # is plotter font
<string>, # font name
<int>, # color
<string>] # text string
```

<Justification> – defined as an integer

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
# 0 : left top
# 1 : left base
# 2 : left bottom
# 3 : center top
# 4 : center base
# 5 : center bottom
# 6 : right top
# 7 : right base
# 8 : right bottom
```

<ImageInfo>:

```
"Image:=", [<RectInfo>,
<ImageData>,
<bool>] # is mirrored
```

<ImageData>:

```
<string>, # file path
<int>, # 0 : use the file path and link to it
# 1 : ignore file path and use next parameter
<string> # text data, only present if preceding int is 1
```

<EditContext>:

<RefPinOption> – defined as an integer

```
# 0 = implied reference to ground
# 1 = single common reference port
# 2 = individual hidden reference pins for each port
# 3 = individual reference pin per port
```

<CompName> – Name of the component defined as a string

<TermAttributes> – An array that includes the following:

```
<Terminal Name> # defined as a string
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<Symbol Pin Name> # defined as a string
<InOut> # defined as an integer>
# 0=in
# 1=out
# 2= inout
```

Export [symbol manager]

Exports symbol(s) to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Symbols** > **Export to Library**		
Parameters	Name	Type	Description
	<NameArray>	Array	Includes the following: *<LibraryName>* – String that defines the name of the library to export the symbol to *<SymbolName>* – String that defines the composite name of the symbol to export
	<LibraryLocation>	String	Location of the library specified with <LibraryName>; options are "`Project`", "`PersonalLib`", or "`UserLib`".
Return Value	None		

Python Syntax	Export([*<LibraryName>*, *<SymbolName>]*, *<LibraryLocation>*)

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`oSymbolManager.Export(["NAME Nexxim Circuit Elements\Distributed\Distributed:bendo:mylib", "mySymbol"], "PersonalLib")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNames [symbol manager]

Returns the names of the symbols (used and unused) in a design. The following script command, **IsUsed,** can then be used to separate used and unused symbols.

UI Access	N/A
Parameters	None
Return Value	An array of strings

Python Syntax	GetNames()

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`symbolNames = oSymbolManager.GetNames()`

IsUsed [symbol manager]

Used to determine if a symbol is used in the design.

UI Access	N/A		
Parameters	Name	Type	Description
	<*SymbolName*>	String	Symbol name to query
Return Value	Boolean; true if the specified symbol is used in the design		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	IsUsed(<*SymbolName*>)

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`IsUsed = oSymbolManager.IsUsed("MySymbol")`

Remove [symbol manager]

Removes a symbol from a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Symbols** > **Remove Symbol**
Parameters	See table below
Return Value	None

Name	Type	Description
<*SymbolName*>	String	Composite name of the symbol to remove
<*IsProjectSymbol*>	Boolean	True if it is a project symbol
<*LibraryName*>	String	Name of the library
<*LibraryLocation*>	String	Location of the library specified with <*LibraryName*>; options are "Project", "PersonalLib", or "UserLib".

Python Syntax	Remove (<SymbolName>, <*IsProjectSymbol*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oSymbolManager.Remove("Nexxim Circuit Elements\Distributed\Distributed:bendo", true, "Project")`

RemoveUnused [symbol manager]

Removes symbols that are not used in the design.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Project** > **Remove Unused Definitions** is similar but operates slightly different and does not record script commands.
Parameters	None
Return Value	Boolean; True if one or more symbols are removed.

Note:

The order of calls to RemoveUnused is significant. As a result, removing definitions in an unordered fashion may cause other symbols in dependent definitions to be rendered unusable.

Also, the symbol and footprint of an unused component are not unusable until after the component itself is removed using the Component Manager Remove script.

Python Syntax	RemoveUnused()
Python Example	`removedSymbols = oSymbolManager.RemoveUnused()`

Technology Manager Script Commands

The technology manager provides access to technologies in a project. The manager object is accessed via the definition manager.

```
oDefinitionManager = oProject.GetDefinitionManager()
oTechnologyManager = oDefinitionManager.GetManager("Technology")
```

The technology manager script commands are listed below.

Add

GetNames

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IsUsed

Remove

Add (Technology Manager)

Adds a technology definition to the library.

<table>
<tr><td>UI Access</td><td>From the Project Manager window, right-click Definitions > Technology and select Add Definition.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><propertyArgs></td><td>Array</td><td>An array that contains these parameters in order:
<Name>
<ModifiedOn>
<Library>
<VLCTech>
<GFD>
<LayerFilter>
<IsLocal></td></tr>
<tr><td><NAME></td><td>String</td><td>Name and extension of the technology file.</td></tr>
<tr><td><ModifiedOn></td><td>Integer</td><td>Optional. An integer corresponding to the number of seconds that elapsed since 00:00 hours, Jan 1, 1970 UTC from the system clock.</td></tr>
<tr><td><Library></td><td>String</td><td>Optional. Identifies a unique definition library.</td></tr>
<tr><td><VLCTech></td><td>String</td><td>Directory path to a technology file.</td></tr>
<tr><td><GFD></td><td>String</td><td>Optional. Directory path to the Green's Function Data file.</td></tr>
<tr><td><LayerFilter></td><td>String</td><td>Optional. Directory path to the layer map file.</td></tr>
<tr><td><IsLocal></td><td>Boolean</td><td>Optional.
True – technology file is saved/stored locally
False – technology file is saved/stored on an external drive</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Add([<NAME:>, <ModifiedOn>, <Library>, <VLCTech>, <LayerFilter>, <IsLocal>])

Python Example	`oTechnologyManager.Add(`
	`  [`
	`  "NAME:Model.vlc.tech",`
	`  "ModifiedOn:=" , 1694708380,`
	`  "Library:=" , "",`
	`  "VLCTech:=" , "C:/raptorx/gds/gpdsimport/Model.vlc.tech",`
	`  "LayerFilter:=" , "",`
	`  "IsLocal:=" , False`
	`  ])`

GetNames (Technology Manager)

Shows the complete list of technologies in the active design's **Technology Library**.

UI Access	From the **Project Manager** window, right-click **Technology** and select **Edit Definition**.
Parameters	None.
Return Value	Array of strings containing the names of identified technology files.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetNames()
Python Example	`oTechnologyManager.GetNames()`

IsUsed (Technology Manager)

Confirms whether a specified technology is applied to the active design.

UI Access	None.
Parameters	Name | Type | Description <*TechName*> | String | Name of technology in the active design's Technology Library.
Return Value	**True** – the specified technology is applied to the active design **False** – the specified technology is not applied to the active design

Python Syntax	IsUsed(<TechName>)
Python Example	`oTechnologyManager.IsUsed("Model.vlc.tech")`

Remove (Technology Manager)

Removes a technology definition from the library.

UI Access	From the **Project Manager** window, right-click **Definitions** > **Technology** and select **Edit Definition**. to open the **Edit Libraries** window. Then select a technology and click **Remove Technology**.
Parameters	Name | Type | Description <*Name*> | String | Name and extension of the technology file. <*ProjectDefinition*> | Boolean | **True** – removes (i.e., deletes) the technology file

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			False – does not remove the technology file
	<LIbraryName>	String	Identifies a unique definition library.
	<LibraryLocation>	String	Directory path to a technology file.
Return Value	None.		

Python Syntax	Remove(<Name>, <ProjectDefinition>, <LibraryName>, <LibraryLocation>)
Python Example	`oTechnologyManager.Remove("Model.vlc.tech", True, "", "Project")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

26 - Definition Editor Script Commands

The Definition Editor controls the use of materials and scripts in a project. Symbol editor script commands and footprint editor script commands are accessed using the Definition Editor.

```
oDefinitionEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "MySymbol")
```

```
oDefinitionEditor = oProject.SetActiveDefinitionEditor("FootprintEditor","MyFootprint")
```

The topics for this section include:

Activate

AddCircuitRefPort

AddLayer

AddPortsToAllNets

AddPortsToNet

AddRefPort

AddStackupLayer

AlignObjects

AlignPorts

AssignCircuitRefPort

AssignRefPort

ChangeLayers

ChangeOptions

ChangeProperty

ClearLayerMappings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClearRefPort

ClearRelative

Connect

ConvertPlanetoTrace

ConvertTracetoPlane

Copy

Create3DStructure

CreateCS

CreateCircle

CreateCircleVoid

CreateCircuitPort

CreateCircuitPortsOnComponents

CreateComponent

CreateEdgePort

CreateGroundPortComponent

CreateGroupSelected

CreateHole

CreateInterfaceGround

CreateInterfacePort

CreateInterfacePortComponent

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateLine

CreateLineFromPolygon

CreateLineVoid

CreateMeasure

CreateNPort

CreateNetClass

CreateObjectFromPolygon

CreatePin

CreatePinGroup

CreatePinGroupPort

CreatePinGroupPortsOnComponents

CreatePolygon

CreatePolygonVoid

CreatePortInstancePorts

CreatePortsOnComponents

CreateRectangle

CreateRectangleVoid

Create Text

CreateTrace

CreateVia

CutOutSubDesign

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeactivateOpen

DeactivateShort

DefeatureObjects

DelNetClass

Delete

DeleteNets

DeletePinGroup

Disconnect

Duplicate

DuplicateAcrossLyrs

Edit

Edit3DComponentDefinition

EraseMeasurements

Expand

ExpandWithPorts

ExportDXF

ExportGDSII

ExportGerber

ExportImage

ExportNCDrill

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportToHFSS

ExportToQ3D

FilterObjectList

FindObjectsByPoint

FindObjectsByPolygon

FlipHorizontal

FlipVertical

GeometryCheckAndAutofix

GetAllLayerNames

GetBBox

GetCSObjects

GetCompInstanceFromRefDes

GetComponentInfo

GetComponentPinInfo

GetComponentPins

GetEditorName

GetLayerInfo

GetMaterialList

GetNetClassNets

GetNetClasses

GetNetConnections

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPolygon

GetPolygonDef

GetPortInfo

GetProperties

GetPropertyValue

GetSelections

Group

Heal

HighlightNet

ImportDXF

ImportGDSII

ModifyNetClass

Move

PageSetup

Paste

Point

Polygon

PositionRelative

PushExcitations

RemoveLayer

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemovePortsFromAllNets

RemovePortsFromNet

RemovePortsOnComponents

Rotate

SelectAll

SetCS

SetLayerMapping

SetNetVisible

StitchLines

Subtract

ToggleViaPin

Ungroup

Unite

UnselectAll

ZoomToFit

Symbol Editor Scripts

Footprint Editor Scripts

AddCircuitRefPort (Layout Editor)

Assigns an edge as a reference to an existing port, modeled as a circuit port.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><NAME:Edge></td><td>String</td><td>Object descriptor header.</td></tr>
<tr><td><Et></td><td>String</td><td>Edge type identifier ("pe" for primitive edge, "pse" for via edge).</td></tr>
<tr><td><Prim></td><td>String</td><td>Name of the primitive (used with "pe").</td></tr>
<tr><td><Edge></td><td>Integer</td><td>Edge number on the primitive.</td></tr>
<tr><td><Sel></td><td>String</td><td>Name of the via to use (used with "pse").</td></tr>
<tr><td><Layer></td><td>Integer</td><td>ID of the layer containing the via pad.</td></tr>
<tr><td><Sx></td><td>Integer</td><td>Start X coordinate of the edge arc.</td></tr>
<tr><td><Sy></td><td>Integer</td><td>Start Y coordinate of the edge arc.</td></tr>
<tr><td><Ex></td><td>Integer</td><td>End X coordinate of the edge arc.</td></tr>
<tr><td><Ey></td><td>Integer</td><td>End Y coordinate of the edge arc.</td></tr>
<tr><td><H></td><td>Integer</td><td>Height of the arc (0 for straight edge).</td></tr>
<tr><td><Rad></td><td>Integer</td><td>Arc size in radians (0 for straight edge).</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	AddCircuitRefPort([<*NAME:Edge*>, [<*Et*>, <*Sel*>, <*Layer*>, <*Sx*>, <*Sy*>, <*Ex*>, <*Ey*>, <*H*>, <*Rad*>]])

Python Example	`oEditor.AddCircuitRefPort(`
	`  [`
	`  "NAME:Edge",`
	`  [`
	`  "Et:=", "pse",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Sel:=", "Via1",
        "Layer:=", 3,
        "Sx:=", "1.0mm",
        "Sy:=", "2.0mm",
        "Ex:=", "3.0mm",
        "Ey:=", "4.0mm",
        "H:=", "0mm",
        "Rad:=", "0"
      ]
    ]
)
```

AddLayer (Layout Editor)

Adds a layer in a layout definition. This command can take all parameters available in ChangeLayers (Layout Editor) but cannot be recorded.

Note: As with other Layout scripting interface commands that modify the layout, this command is not intended for use within scripts that define footprints. The command behavior from within such a script is undefined and may be unexpected. Use the LayoutHost scripting interface commands within scripts that define footprints.

UI Access	None		
Parameters	Name	Type	Description
	<LayerArray>	Array	An array that contains these parameters in order: • <Layername>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <Type> • <TopBottom> • <Color> • <Transparency> • <Pattern> • <VisFlag> • <Locked> • <Draw>
<LayerName>	String	Name of layer.
<Type>	String	Type of layer, such as "user", "soldermask", "solderpaste", "silkscreen", "assembly", "wirebond", "glue", "user", "Postprocessing", or "outline".
<TopBottom>	String	Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
<Color>	Integer	Optional. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
<Transparency>	Integer	Optional. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
	<Locked>	Boolean	Optional. Whether the layer is locked.
	<Draw>	Integer	Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
Return Value	None		

Python Syntax	oEditor.AddLayer(["NAME:layers", "Name:=", "<LayerName>", "Type:=", "<Type>", "Top Bottom:=", "<TopBottom>", "Color:=", <Color>, "Transparency:=", <Transparency>, "Pattern:=", <Pattern>, " VisFlag:=" , <VisFlag>, "Locked:=", <Locked>, "DrawOverride:=", <Draw>,])
Python Example	`oEditor = oDesign.SetActiveEditor("Layout")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	oEditor.AddLayer(["NAME:layers", "Name:=","signal", "Type:=", "assembly", "Top Bottom:=", "neither", "Color:=", 16711680, "Transparency:=", 0, "Pattern:=", 1, "VisFlag:=", 127, "Locked:=", False, "DrawOverride:=", 0,])

AddPortsToAllNets (Layout Editor)

Adds ports to all the pins on all nets.

UI Access	From the **Layout** tab > **Nets**, right-click and click **Create Ports**.
Parameters	None
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddPortsToAllNets()
Python Example	`oEditor.AddPortsToAllNets()`

AddPortsToNet (Layout Editor)

Adds the selected ports to the current selected net.

UI Access	Right-click within the **Layout Editor** and select **Port** > **Create Ports on Net**.
Parameters	See table below
Return Value	None

Name	Type	Description
<Parameters>	Array	Structured array containing these parameters in order: • <NAME:Nets> • <net_name>
<NAME:Nets>	String	Required first parameter which must have the specified value (i.e., "Nets").
<port_name>	String	A comma-separated list of port names (e.g., "port1", "port2", "port3", et cetera).

Python Syntax	AddPortsToNet([*<NAME:Nets>, <net_name>*])

Python Example	`oEditor.AddPortsToNet(`
	`  [`
	`  "NAME:Nets",`
	`  "port1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"port2"])

AddRefPort (Layout Editor)

Creates a reference port from edges and associates the reference port with each of the named ports.

UI Access	Right-click within the **Layout Editor** and select **Port** > **Create**.
Parameters	See table below

Name	Type	Description
<*Port_List*>	Array	Structured array containing one or more *Port_Name* strings. Each *Port_Name* represents a named port (e.g., "port1", "port2", "port3").
<*Port_Name*>	String	Individual port name string used within the *Port_List* array.
<*Edge_Array*>	Array	Structured array containing these parameters in order: • <NAME:Contents> — Required string with value "Contents" • <Edge> — One entry per *Port_Name* in *Port_List* Each *Edge* entry may reference a predefined edge object (e.g., "edge_description") or contain a full *Edge_Description* array inline.
<*NAME:Contents*>	String	Required first parameter which must have the specified value (i.e., "Contents").
<*Edge*>	Array	Structured array containing an <Edge_description> parameter for each <Port_Name> parameter provided in the <Port List> array.
<*Edge_ Description*>	Array	Name of the array containing the geometry of the selected edge, containing these parameters in order:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Layer> • <Port> • <Et> • <Pe> • <Prim> • <Edge> • <Edge#> Via edges: • <Et> • <Pse> • <Sel> • <Via> • <Layer> • <Layer_ID> • <Sx> • <Sy> • <Ex> • <Ey> • <H> • <Rad>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Port>	String	Name of the selected port or pin.
	<Et>	String	Specifies the edge type: Pe for primitive edge or Pse for via edge.
	<Prim>	String	Specifies the primitive type associated with the edge.
	<Edge#>	Integer	Index number of the selected primitive.
	<Sel>	Integer	Selection index for the via. Used to identify which via instance is associated with the edge. Typically set to 0 unless multiple vias are present.
	<Via>	String	Name of the selected via.
	<Layer>	Integer	The ID of the layer of the pad of the via to use.
	<Sx>	Integer	X coordinate of the start point of the edge arc.
	<Sy>	Integer	Y coordinate of the start point of the edge arc.
	<Ex>	Integer	X coordinate of the end point of the edge arc.
	<Ey>	Integer	Y coordinate of the end point of the edge arc.
	<H>	Integer	Height of the edge arc (0 for a straight edge).
	<Rad>	Integer	Size of the edge arc in radians (0 for a straight edge).
Return Value	None		

Python Syntax	AddRefPort([<port_list>], [<edge_array>])

Python Example	`oEditor.AddRefPort([`
	`  "port1",`
	`  "port2"],`
	`  ["NAME:Contents",`
	`  "edge:=" , "edge_description",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`"edge:=" , "edge_description"]`)

AlignObjects (Layout Editor)

Aligns objects (e.g. primitives, components, et cetera) relative to each other.

UI Access	From **Draw**, select **Align**.
Parameters	See table below

Name	Type	Description
<align_ parameters>	Array	Structured array containing these parameters in order: • <NAME:align> • <horz_align> • <vert_alignment
<NAME:align>	String	Required first parameter which must have the specified value (i.e., "align").
<horz_align>	String	***Optional.*** Must be any one of the following: • Leave Blank — no horizontal alignment. • AlignLeft — align objects left. • AlignRight — align objects right. • CenterHorz — align object centers horizontally. • DistributeCentersHorz — places objects so the horizontal spaces between their centers are equal . • SpaceEdgesHorz — places objects so the horizontal spaces between their edges are equal
<vert_align>	String	***Optional.*** Must be any one of the following: • Leave Blank — no vertical alignment.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• AlignTop — align tops of objects. • AlignBottom — align bottoms of objects. • CenterVert — align object centers vertically. • DistributeCentersVert — places objects so the vertical spaces between their centers are equal . • SpaceEdgesVert — places objects so the vertical spaces between their edges are equal .
	<Element_Parameters>	Array	Structured array containing these parameters in order: • <NAME:elements> • <elements>
	<NAME:elements>	String	Required first parameter which must have the specified value (i.e., "elements").
	<elements>	String	A comma-separated list of object names (e.g., "polygon_01", "polygon_02", "polygon_03", et cetera).
Return Value	None		

Python Syntax	AlignObjects([*<NAME:align>, <horz_align>, <vert_align>*], [<NAME:elements>, <elements>])

Python Example	`oEditor.AlignObjects(`
	`  [`
	`  "NAME:align",`
	`  "AlignLeft",`
	`  "AlignTop"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	```], ["NAME:elements", "polygon_01", "polygon_02"]) ```

AlignPorts (Layout Editor)

Causes automatic alignment of the microwave ports of the components, edge ports, and pins referenced in the argument. If the list is empty, aligns all the microwave ports in the layout.

UI Access	From **Draw**, select **Align MW Ports**.		
Parameters	Name	Type	Description
	<*align_parameters*>	Array	Structured array containing these parameters in order: • <NAME:elements> • <object_name>
	<*NAME:elements*>	String	Required first parameter which must have the specified value (i.e., "elements").
	<*port_name*>	String	A comma-separated list of port names (e.g., "port1", "port2", "port3", et cetera).
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AlignPorts(<*NAME:elements*>, <*port_name*>)

Python Example	`oEditor.AlignPorts(`
	`  ["NAME:Elements",`
	`  "port1",`
	`  "port2"`
	`  ])`

AssignCircuitRefPort (Layout Editor)

Assigns the internal port as a reference to an existing port (i.e., model as a circuit port).

UI Access	None		
Parameters	Name	Type	Description
	<*Port List*>	Array	Structured array containing a comma-separated list of any number of <Port_Name> parameters (e.g., "port1", "port2", "port3", et cetera).
	<*Via*>	String	Name of the selected via.
	<*Layer*>	Integer	The ID of the layer of the pad of the via to use.
	<*Sx*>	Integer	X coordinate of the start point of the edge arc.
	<*Sy*>	Integer	Y coordinate of the start point of the edge arc.
	<*Ex*>	Integer	X coordinate of the end point of the edge arc.
	<*Ey*>	Integer	Y coordinate of the end point of the edge arc.
	<*H*>	Integer	Height of the edge arc (0 for a straight edge).
	<*Rad*>	Integer	Size of the edge arc in radians (0 for a straight edge).
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddRefPort([<port_list>], [<edge_array>])

Python Example	`oEditor.AddRefPort([`
	`  "port1",`
	`  "port2"],`
	`  ["NAME:Contents",`
	`  "edge:=", ["edge_description"],`
	`  "edge:=", ["edge_description"]`
	`])`

AssignRefPort (Layout Editor and IC Mode)

Assigns a named internal port as a reference for each of the specified ports.

UI Access	None		
Parameters	Name	Type	Description
	<Port_List>	Array	Array of port names to be assigned a common reference port (e.g., "`port1`", "`port2`").
	<Reference_ Edge_Array>	Array	Structured array counting the following parameters in order: • `<NAME:Contents>` • One `<Edge>` entry per port in *Port_List*. Each `Edge` entry contains either a primitive or via edge description.
	<NAME:Contents>	String	Object descriptor header.
	<Edge>	Array	Structured array counting the following parameters in order: • `<Et>` • `<Prim>` or `<Sel>` • `<Edge#>` (if primitive)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Layer> • <Layer_ID> (if via) • <Sx, Sy> • <Ex, Ey> • <H> • <Rad> Each entry describes either a primitive or via edge.
	<Layer>	String	Name of the layer containing the edge or via.
	<Port>	String	Name of the port or pin to assign as reference.
	<Et>	String	Specifies the edge type: "Pe" for primitive edge or "Pse" for via edge.
	<Prim>	String	Name of the primitive associated with the edge.
	<Edge#>	Integer	Index number of the edge on the primitive.
	<Sel>	String	Name of the via to use for via edge assignment.
	<Layer_ID>	Integer	ID of the layer containing the pad of the via.
	<Sx>	Integer	X coordinate of the start point of the edge arc.
	<Sy>	Integer	Y coordinate of the start point of the edge arc.
	<Ex>	Integer	X coordinate of the end point of the edge arc.
	<Ey>	Integer	Y coordinate of the end point of the edge arc.
	<H>	Integer	Height of the edge arc (0 for a straight edge).
	<Rad>	Integer	Arc size in radians (0 for a straight edge).
Return Value	None		

Python Syntax	AssignRefPort([*Port_List*], [*Reference_Edge_Array*])
Python Example	`oEditor.AssignRefPort(` `  ["port1", "port2"],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
"NAME:Contents",
"edge:=", [
  "Layer:=", "TopLayer",
  "Port:=", "port1",
  "Et:=", "Pe",
  "Prim:=", "trace1",
  "Edge#:=", 1
],
"edge:=", [
  "Et:=", "Pse",
  "Sel:=", "via1",
  "Layer:=", "BottomLayer",
  "Layer_ID:=", 3,
  "Sx:=", 100.0,
  "Sy:=", 200.0,
  "Ex:=", 300.0,
  "Ey:=", 400.0,
  "H:=", 0.0,
  "Rad:=", 0.0
]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	)

ChangeProperty

Changes the properties of an object in the history tree.

UI Access	Right-click an object in the History Tree and select **Properties**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*propertyArgs*></td><td>Array</td><td>Structured array. The properties vary depending on the object. Due to the number of potential configurations, it is recommended that you generate this script using the UI's **Automation** tab.</td></tr></table>
Return Value	None.

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	**Example: Changing the Position of a Box**
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DCmdTab",`
	`  ["NAME:PropServers" , "Box1:CreateBox:1" ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Position", "X:=" , "0.35in", "Y:=" , "0.55in", "Z:=" , "0in"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          ]
        ]
      ]
    ])
```

Example: Offset the Position of a 3D Component or Layout Component with a Variable

```
oDesign.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:LocalVariableTab",
      ["NAME:PropServers",
        "LocalVariables"
      ],
      ["NAME:NewProps",
        ["NAME:zH",
        "PropType:=" , "VariableProp",
        "UserDef:=" , True,
        "Value:=" , "50mm"
        ]
      ]
    ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers",
        "LC1_1"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      ],
      ["NAME:ChangedProps",
        ["NAME:Position",
        "X:=" , "0mm",
        "Y:=" , "0mm",
        "Z:=" , "zH"
        ]
      ]
    ]
  ])
```

Example: Changing a Box's Material and Wireframe Display

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers" , "Box1" ],
       ["NAME:ChangedProps",
       ["NAME:Material", "Value:=" , "\"vacuum\"" ],
       ["NAME:Display Wireframe", "Value:=" , True ]
      ]
    ]
  ])
```

Example: Change Default Handle (reference coordinate system) for a 3D Component or Layout Component

```
oEditor.ChangeProperty(
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   ["NAME:AllTabs",
     ["NAME:General",
       ["NAME:PropServers","LC1_1"],
       ["NAME:ChangedProps",
         ["NAME:Handle","Value:=" , " NotchOutY"]
       ]
     ]
   ])

oEditor.ChangeProperty(
   ["NAME:AllTabs",
      ["NAME:Maxwell2D",
        ["NAME:PropServers", "Design Settings" ],
       ["NAME:ChangedProps",
         ["NAME:Model settings/Depth", "Value:=" , "2.4mm" ]
        ]
     ]
   ])
```

ClearLayerMappings (Layout Editor)

Clears all layer mappings associated with the specified component.

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><ComponentName></td><td>String</td><td>Name of the component whose layer mappings will be cleared.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ClearLayerMappings(<componentName>)

Python Example	`ClearLayerMappings(`
	`  "Component_A"`
	`)`

ClearRefPort (Layout Editor)

Clears reference ports from each of the specified named ports.

UI Access	From **Draw** > **Clear Reference Port**.		
Parameters	Name	Type	Description
	<PortList>	Array	Array of port names from which reference ports will be cleared.
Return Value	None		

Python Syntax	ClearRefPort(<PortList>)

Python Example	`ClearRefPort(`
	`  [ "Port1", "Port2", "Port3" ]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClearRelative (Layout Editor)

Clears the relative positioning between a coordinate system (CS) and another object.

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "elements").</td></tr><tr><td><*CS_name*></td><td>String</td><td>Name of the coordinate system whose relative positioning is to be cleared.</td></tr></table>
Return Value	None

Python Syntax	ClearRelative(<*NAME:elements*>, <*CS_name*>)

Python Example	`oEditor.ClearRelative(`
	`  ["NAME:Elements",`
	`  "CS1",`
	`  "CS2"`
	`  ])`

Connect (Layout Editor)

Connects the referenced objects. After execution, the objects will belong to the same electrical net.

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "ele-</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<table><tr><td></td><td></td><td>ments").</td></tr><tr><td><*object_name*></td><td>String</td><td>Name of the object to connect. Multiple objects may be listed.</td></tr></table>
Return Value	None

Python Syntax	Connect(<*NAME:elements*>, <*object_name*>, <*object_name*>, …)

Python Example	`oEditor.Connect(`
	`  ["NAME:Elements",`
	`  "rect_01",`
	`  "rect_02"`
	`  ])`

ConvertPlaneToTrace (Layout Editor)

Converts selected polygon planes into trace objects.

UI Access	Right-click a polygon, then select **Draw** > **Planes to Traces**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Polygon_Data*></td><td>Array</td><td>List of polygon objects to convert. Each entry should reference a valid polygon name.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ConvertPlaneToTrace(<*Polygon_Data*>)

Python Example	`oEditor.ConvertPlaneToTrace(`
	`  ["polygon_01",`
	`  "polygon_02"`
	`  ])`

ConvertTraceToPlane (Layout Editor)

Converts selected trace paths into polygon plane objects.

UI Access	Right-click a path, then select **Draw** > **Traces to Planes**.		
Parameters	Name	Type	Description
	<*Line_Data*>	Array	List of trace path objects to convert. Each entry should reference a valid line or path name.
Return Value	None		

Python Syntax	ConvertTraceToPlane(<*Line_Data*>)

Python Example	`oEditor.ConvertTraceToPlane(`
	`  ["trace_01",`
	`  "trace_02"`
	`  ])`

Copy (Layout Editor)

Use: Copies the referenced objects to the clipboard.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: **Copy** Array (<object_name>, // 1st object

<object_name>, // 2nd object

Return Value: Returns the name of the new net.

CopyToHFSS 3D Layout (Layout Editor)

Use: Generate a single, flat, HFSS 3D Layout design from a selection of components and/or sub-circuits.

Command: CopyToHFSS 3D Layout

Syntax: CopyToHFSS 3D Layout Array("NAME:elements", <"obj1">, <"obj2">, …)

Return Value: None

Parameters: <"obj1">

Type: text

// component or sub-circuit instance name.

Create3DStructure (Layout Editor)

Place a set of primitives into a 3D structure coordinate system (i.e., a 3D via or cross-layer plate).

UI Access	None		
Parameters	Name	Type	Description
	<3DStructureName>	String	Name of the 3D structure.
	<NAME:Elements>	String	Object descriptor header.
	<ElementN	String	Names of layout primitives to include in the structure, sep-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			arated by commas.
Return Value	None		

Python Syntax	Create3DStructure(<3DStructureName>, [<NAME:Elements>, <Element1>, <Element2>, …])

Python Example	`oEditor.Create3DStructure(`
	`  "ViaStructure", [`
	`  "NAME:Elements",`
	`  "ViaTop",`
	`  "ViaBarrel",`
	`  "ViaBottom"`
	`])`

CreateCS (Layout Editor)

Inserts a new coordinate system (CS) into the currently active Layout design.

UI Access	From **Draw**, select **Coordinate System** > **Create**.		
Parameters	Name	Type	Description
	<*NAME:Contents*>	String	Required header string identifying the parameter block.
	<*Full_Definition*>	Array	Contains: • `Type`: Type of the coordinate system (e.g., "CS") • `N`: Name of the coordinate system
	<*Type*>	String	Type of the coordinate system.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Name	Type	Description
<N>	String	Name of the coordinate system.
<Placement>	Array	Contains: • `X`: X origin coordinate • `Y`: Y origin coordinate
<X>	Double	X origin coordinate.
<Y>	Double	Y origin coordinate.
<Clf>	Boolean	True if a color is being specified by the next parameter.
<Col>	Integer	Color index for the coordinate system. Only used if `Clf` is true.
<Name>	String	Requested name for the coordinate system.
<RelPos>	Array	Contains: • `t`: Edge type ("pe" for primitive, "pse" for via) • `from`: Primitive or via name • `pos`: Position along the edge (0–1) • `et`: Edge type ("pe" or "pse") • `prim`: Primitive name (if using primitive edge) • `edge`: Edge number (if using primitive edge) • `sel`: Via name (if using via edge) • `layer`: Layer ID (if using via edge) • `sx`, `sy`: Start X, Y of arc (via edge) • `ex`, `ey`: End X, Y of arc (via edge) • `h`: Arc height • `rad`: Arc size in radians
<t>	String	Edge type: "pe" for primitive, "pse" for via.
<from>	String	Name of the primitive or via.
<pos>	Double	Position along the edge (0 = start, 1 = end).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Name	Type	Description
<et>	String	Edge type repeated: "pe" or "pse".
<prim>	String	Primitive name (used for primitive edge).
<edge>	Integer	Edge number on the primitive.
<sel>	String	Via name (used for via edge).
<layer>	Integer	Layer ID of the via pad.
<sx>, *<sy>*	Double	X, Y location of the start point of the arc.
<ex>, *<ey>*	Double	X, Y location of the end point of the arc.
<h>	Double	Height of the arc (0 for straight edge).
<rad>	Double	Arc size in radians (0 for straight edge).

Return Value	Returns the name of the created relative coordinate system.

Python Syntax	CreateCS([<NAME:Contents>, <Full_Definition>, <Placement>, <Clf>, <Col>, <Name>, <RelPos>])
Python Example	

```
oEditor.CreateCS([
  "NAME:Contents",
  "Full_Definition:=", ["Type:=", "CS", "N:=", "MyCS"],
  "Placement:=", ["X:=", 0.0, "Y:=", 0.0],
  "Clf:=", True,
  "Col:=", 3,
  "Name:=", "MyCS",
  "RelPos:=", [
  "t:=", "pe",
  "from:=", "Prim1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   "pos:=", 0.5,
   "et:=", "pe",
   "prim:=", "Prim1",
   "edge:=", 2
   ]
])
```

CreateCircle (Layout Editor)

Creates a circle primitive object and adds it to the current layout.

UI Access	None
Parameters	See table below
Return Value	Object name (string)

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*CircleGeometry*>	Array	Array of geometric properties for the circle.
<*LayerName*>	String	Name of the layer.
<*Lw*>	Integer	Border line width.
<*X*>	Integer	Center X coordinate.
<*Y*>	Integer	Center Y coordinate.
<*R*>	Integer	Radius.

Python Syntax	CreateCircle([<*NAME:Contents*>, <*CircleGeometry*>, [<*LayerName*>, <*Lw*>, <*X*>, <*Y*>, <*R*>]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.CreateCircle(`
	`  [`
	`  "NAME:Contents",`
	`  "CircleGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "Lw:=", "0.2mm",`
	`  "X:=", "10mm",`
	`  "Y:=", "15mm",`
	`  "R:=", "5mm"`
	`  ]`
	`  ]`
	`)`

CreateCircleVoid (Layout Editor)

Creates a circle void and adds it to a particular parent primitive.

UI Access	Available only through scripting.
Parameters	

Name	Type	Description
<NAME:Contents>	String	Header string identifying the array contents.
<Owner>	String	Name of the parent primitive object to which the void is added.
<Circle VoidGeometry>	Array	Contains geometric parameters for the void. Format is assumed to match `CreateCircle` geometry array.
<*LayerName*>	String	Name of the layer.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Lw>	Integer	Border line width.
	<X>	Integer	Center X coordinate.
	<Y>	Integer	Center Y coordinate.
	<R>	Integer	Radius
Return Value	Void object name.		

Python Syntax	oEditor.CreateCircleVoid([<NAME:Contents>, <Owner>, <Circle VoidGeometry>])

Python Example	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.CreateCircleVoid(`
	`  ["NAME:Contents",`
	`  "Owner:=", "Pad1",`
	`  "Circle VoidGeometry:=",`
	`  ["LayerName:=", "Top",`
	`  "lw:=", "0.1mm",`
	`  "x:=", "0mm",`
	`  "y:=", "0mm",`
	`  "r:=", "0.5mm"]`
	`  ]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateCircuitPort (Layout Editor)

Creates a circuit port between two points.

UI Access	From **Draw**, select **Create Circuit Ports**.
Parameters	

Name	Type	Description
<*NAME:Location*>	String	Object descriptor header.
<*PosLayer*>	String	Positive terminal layer name.
<*X0*>	Integer	X coordinate of the positive terminal.
<*Y0*>	Integer	Y coordinate of the positive terminal.
<*NegLayer*>	String	Negative terminal layer name.
<*X1*>	Integer	X coordinate of the negative terminal.
<*Y1*>	Integer	Y coordinate of the negative terminal.

Return Value	None
Python Syntax	CreateCircuitPort([<*NAME:Location*>, <*PosLayer*>, <*X0*>, <*Y0*>, <*NegLayer*>, <*X1*>, <*Y1*>])
Python Example	

```
oEditor.CreateCircuitPort(
  [
       "NAME:Location",
       "PosLayer:=", "TopLayer",
       "X0:=", "5mm",
       "Y0:=", "10mm",
       "NegLayer:=", "BottomLayer",
       "X1:=", "5mm",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Y1:=", "0mm"
    ]
)
```

CreateCircuitPortsOnComponents (Layout Editor)

Creates circuit ports on a selected component.

UI Access	After selecting a component, right-click within the **Layout Editor** and select **Port** > **Create Ports on Component**.
Parameters	

Name	Type	Description
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:Components> • <ComponentName>
<*NAME:Components*>	String	Required first parameter which must have the specified value (i.e., "Components").
<*ComponentName*>	String	Names of the target components, separated by commas.
<*Parameters*>	Array	Structured array containing these parameters in order: • <NAME:Nets> • <NetNames>
<*NAME:Nets*>	String	Required first parameter which must have the specified value (i.e., "Nets")
<*NetNames*>	String	Names of the nets to place ports on, separated by commas.
<*ReferenceNet*>	String	Name of the net to use as a reference.

Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateCircuitPortsOnComponents ([<NAME:Components>,<ComponentName>], [<NAME:Nets>, <NetNames>], <ReferenceNet>)

Python Example	`oEditor.CreateCircuitPortsOnComponents(`
	`[`
	`"NAME:Components",`
	`U0`
	`],`
	`["NAME:Nets",`
	`"net3",`
	`"net4"`
	`], "net5"`
	`])`

CreateComponent (Layout Editor)

Places a component in the layout editor.

UI Access	None
Parameters	

Name	Type	Description
<*NAME:Contents*>	String	Required header string identifying the parameter block.
<*Definition_Name*>	String	Name of the component definition to place.
<*Placement*>	Array	Structured array containing: `X`, `Y`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Name	Type	Description
<X>	Double	X coordinate of the component origin.
<Y>	Double	Y coordinate of the component origin.
<Layer>	String	Name of the layer on which to place the component.
<StackupLayers>	Array	Optional array of layer mappings. Each entry contains: `FootprintLayer`, `DesignLayer`
<FootprintLayer>	String	Name of the footprint stackup layer.
<DesignLayer>	String	Name of the corresponding design stackup layer.
<DrawLayers>	Array	Optional array of drawing layer mappings. Each entry contains: `FootprintLayer`, `DesignLayer`
<FootprintLayer>	String	Name of the footprint drawing layer.
<DesignLayer>	String	Name of the corresponding design drawing layer.

Return Value	Returns the name of the newly created component.

Note: Each Layer mapping is specified as a single quoted string in the format `"<footprint layer>:<design layer>"`. If a layer does not map to anything, use `"0"` for the unmapped side (e.g., `"Measures:0"` — footprint layer "Measures" has no design layer mapping, or `"0:Dielectric"` — design layer "Dielectric" has no footprint layer mapping.

Note: The `StackupLayers` and `DrawLayers` arguments are optional. If omitted, mappings are determined automatically.

Python Syntax	CreateComponent([<NAME:Contents>, <Definition_Name>, <Placement>, <Layer>, <StackupLayers>, <DrawLayers>])
Python Example	`oEditor.CreateComponent([`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "NAME:Contents",
  "Definition_Name:=", "MyComponent",
  "Placement:=", ["X:=", 0.0, "Y:=", 0.0],
  "Layer:=", "TopLayer",
  "StackupLayers:=", ["Top:Dielectric", "Measures:0"],
  "DrawLayers:=", ["Silkscreen:Top", "0:Dielectric"]
])
```

CreateEdgePort (Layout Editor)

Creates an edge port using the specified edges.

UI Access	**Draw** > **Port** > **Create** or **Right-click** > **Port** > **Create**.
Parameters	

Name	Type	Description
<*EdgePortArray*>	Array	Identifies the contents of the array (i.e., "Contents") containing the following arguments: <NAME:Contents> <Edge> <External>
<*NAME:Contents*>	String	Identifies the contents of the array (i.e., "Contents").
<*Edge*>	Array	Array containing separate arrays for primitive edges and via edges. For primitive edges, array contains: <Et>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		<Prim> <Edge> For via edges, array contains: <Et> <Sel> <Layers> <Sx> <Sy> <Ex> <Ey> <H> <Rad> **Note:** Array can repeat in the same command.
<*Et*>	String	Type of new edge port (i.e., "pe" for primitive edge or "pse" for pad-stack edge or via).
<*Prim*>	String	Name of the target primitive.
<Edge#>	Integer	Edge number of the target primitive.
<Sel>	String	Name of the target via.
<Layers>	Integer	Layer ID of the pad of the target via.
<Sx>	Integer	Starting X coordinate for the arc of the new edge port.
<Sy>	Integer	Starting Y coordinate for the arc of the new edge port.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Ex>	Integer	Ending X coordinate for the arc of the new edge port.
	<Ey>	Integer	Ending Y coordinate for the arc of the new edge port.
	<H>	Integer	Arc height of the new edge port (e.g., enter 0 for a straight edge).
	<rad>	String	Arc size of the new edge port (e.g., enter 0 for a straight edge).
	<External>	Boolean	True — the port is external. False — the port is internal.
Return Value	Text containing the name of the created edge port. (Returns an empty name if the edge port is not created.)		

Python Syntax	CreateEdgePort(<*EdgePortArray*>)

Python Example	Primitive edge port:
	`oEditor.CreateEdgePort (("NAME:Contents", "edge:=", ("et:=", "pe",`
	`"prim:=", "rect_167", "edge:=", 0), "edge:=", ("et:=", "pe", "prim:=",`
	`"rect_167", "edge:=", 3), "external:=", true))`
	Via edge port:
	`oEditor.CreateEdgePort (("NAME:Contents", "edge:=", ("et:=", "pse",`
	`"sel:=", "via_0", "layer:=", 10, "sx:=", -0.0015, "sy:=", -0.0015, "ex:=", 0.0015,`
	`"ey:=", -0.0015, "h:=", 0, "rad:=", 0), "external:=", true))`

sure

CreateGroundPortComponent (Layout Editor)

Creates a ground port component at unconnected pins of the selected port.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	From the **Layout Editor**, right-click a port and select **Port** > **Nexxim Ports** > **Add Grounds at Unconnected Pins**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Component_Data*></td><td>Array</td><td>A structured array consisting of ground port data.</td></tr></table>
Return Value	None

Python Syntax	oEditor.CreateGroundPortComponent ([<Component_Data>])
Python Example	`oEditor.CreateGroundPortComponent(Component_Data)`

CreateGroupSelected (Layout Editor)

Groups a set of layout primitives into a subdesign.

UI Access	From **Layout Editor**, navigate to to **Layout** > **Group into Subdesign**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME:elements*></td><td>String</td><td>Required header string identifying the parameter block.</td></tr><tr><td><*Selectable Name*></td><td>String</td><td>Name of each primitive to include in the group. Multiple entries allowed.</td></tr></table>
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateGroupSelected([*elements*])

Python Example	`oEditor.CreateGroupSelected([`
	`  "NAME:elements",`
	`  "Selectable Name:=", "Resistor1",`
	`  "Selectable Name:=", "Capacitor2"`
	`])`

CreateHole (Layout Editor)

Places a Hole object.

UI Access	From **Layout Editor**, navigate to **Draw** and select **Hole**.

Parameters

Name	Type	Description
<Contents>	Array	Structured array containing the full hole definition and all associated parameters.
<NAME:Full_Definition>	Array	Structured array containing hole type, properties, layer range, and placement.
<Type>	String	Must be set to "hole".
<Properties>	Array	Structured array of hole geometry parameters: <VariableProp> entries for radius and sides.
<VariableProp>	Array	Each entry defines a named property: ("Radius", "D", "", <Value>) or ("Sides", "D", "", <iInteger>).
<From_Layer>	String	Name of the starting layer for the hole.
<To_Layer>	String	Name of the ending layer for the hole.
<Placement>	Array	Structured array defining the hole position: <X>, <Y> coordinates.
<X>	Integer	X coordinate of the hole center.
<Y>	Integer	Y coordinate of the hole center.
<Layer>	String	Layer name used for placement reference.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Returns the name of the newly created Hole.

Python Syntax	CreateHole([<Contents>])

Python Example	`oEditor.CreateHole([`
	`  "NAME:Contents",`
	`  ["NAME:Full_Definition",`
	`  "Type:=", "hole",`
	`  ["NAME:Properties",`
	`  "VariableProp:=", ["radius", "D", "", 1.5],`
	`  "VariableProp:=", ["sides", "D", "", 6]],`
	`  "From_Layer:=", "Top",`
	`  "To_Layer:=", "Bottom",`
	`  "Placement:=", ["x:=", 10, "y:=", 20],`
	`  "Layer:=", "Top"]`
	`])`

CreateInterfaceGround (Layout Editor)

Use: Create an interrface ground

Command: Right-click selected port > Port > Nexxim Ports > Circuit Ground

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None

Parameters: <Port_Data>

Type: array

CreateInterfacePort (Layout Editor)

Use: Create an interrface port

Command: Right-click selected port > Port > Nexxim Ports > Circuit Interface

Syntax: CreateInterfacePort <Port_Data>

Parameters: <Port_Data>

Type: array

CreateInterfacePortComponent (Layout Editor)

Use: Create an interface port component

Command: Right-click selected port > Port > Nexxim Ports > Add Interface Ports at Unconnected Pins

Syntax: CreateInterfacePortComponent <Port_Data>

Parameters: <Component_Data>

Type: array

CreateLine (Layout Editor)

Creates a line primitive object.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*LineGeometry*>	Array	Array of geometric properties for the line.
	<*LayerName*>	String	Name of the layer.
	<*Lw*>	Integer	Line width.
	<*EndStyle*>	Integer	Line end style: 0 = flat end, 1 = extended end, 2 = round end.
	<*JoinStyle*>	Integer	Line join style: 0 = unmitered join, 1 = mitered join, 2 = optimally mitered join, 3 = radially mitered join, 4 = chord mitered join.
	<*VertexSequence*>	Array	Sequence of vertex coordinates: "n:=" count, followed by "x0:=", "y0:=", "x1:=", "y1:=", etc.
Return Value	Object name (string)		

Python Syntax	CreateLine([<*NAME:Contents*>, <*LineGeometry*>, [<*LayerName*>, <*Lw*>, <*EndStyle*>, <*JoinStyle*>, <*VertexSequence*>]])

Python Example	`oEditor.CreateLine(`
	`  [`
	`  "NAME:Contents",`
	`  "LineGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "Lw:=", 2,`
	`  "EndStyle:=", 1,`
	`  "JoinStyle:=", 2,`
	`  "n:=", 2,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"x0:=", 10, "y0:=", 15, "x1:=", 25, "y1:=", 30]])

CreateLineFromPolygon (Layout Editor)

Use: Create a line/trace object on the specified layer and net from the provided Polygon object. Net is optional.

Syntax: oLayout.CreateLineFromPolygon(<oPolygon>, <width>, <bendType>, <startCapType>, <endCapType>, <Layer>, <Net>)

Parameters: width - line width, in meters

bendType - {'corner', 'round'}

startCapType, endCapType - {'flat', 'extended', 'round'}

CreateLineVoid (Layout Editor)

Creates a line void and adds it to a specified parent primitive.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.
	<Owner>	String	Name of the parent primitive.
	<Line voidGeo-	Array	Array defining the line void geometry by two pairs of X,Y

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	metry>		coordinates.
	<*X1*>	Integer	Start X coordinate.
	<*Y1*>	Integer	Start Y coordinate.
	<*X2*>	Integer	End X coordinate.
	<*Y2*>	Integer	End Y coordinate.
Return Value	Void object name.		

Python Syntax	CreateLineVoid([<*NAME:Contents*>, <*Owner*>, <*Line VoidGeometry*>, [<*X1*>, <*Y1*>, <*X2*>, <*Y2*>]])

Python Example	`oEditor.CreateLineVoid(`
	`  [`
	`  "NAME:Contents",`
	`  "Line VoidGeometry:=", [`
	`  "X1:=", "0mm",`
	`  "Y1:=", "0mm",`
	`  "X2:=", "10mm",`
	`  "Y2:=", "0mm"`
	`  ]`
	`  ]`
	`)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateMeasure (Layout Editor)

Creates a measurement.

UI Access	None.		
Parameters	*<Measure_Array>*	Array	An array consisting of the following parameters in order: • *<NAME:Contents>* • *<MeasurementGeometry>* • *<text_style>*
	<NAME:Contents>	String	Identifies the contents of the measurement definition array.
	<Meas-urementGeometry>	Array	Array containing the measurement geometry parameters in order: • *<LayerName>* • *<Lw>* • *<Sx>* • *<Sy>* • *<Ex>* • *<Ey>*
	<LayerName>	String	Specifies the layer on which the measurement geometry is drawn.
	<Lw>	Double	Line width of the measurement geometry.
	<Sx>	Double	X-coordinate of the measurement start point.
	<Sy>	Double	Y-coordinate of the measurement start point.
	<Ex>	Double	X-coordinate of the measurement end point.
	<Ey>	Double	Y-coordinate of the measurement end point.
	<text_style>	Array	Text style parameters in order: • *<Name>*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• *<IsPlot>* • *<Font>* • *<Size>* • *<Angle>* • *<Weight>* • *<Just>* • *<Mirror>* • *<Scales>*
	<Name>	String	Name assigned to the measurement text object.
	<IsPlot>	Boolean	Indicates whether the measurement text is plotted.
	<Font>	String	Font used for the measurement text.
	<Size>	Double	Font size of the measurement text.
	<Angle>	Double	Rotation angle of the text in degrees.
	<Weight>	String	Text weight (i.e., Normal, Bold).
	<Just>	String	Text justification (i.e., Left, Center, Right).
	<Mirror>	Boolean	Mirrors the text horizontally when True.
	<Scales>	Boolean	Determines whether the text scales with zoom level.
Return Value	Name of the measured object.		

Python Syntax	oEditor.CreateMeasure([<NAME:Contents>, <MeasurementGeometry>, <text_style>])
Python	`oEditor.CreateMeasure(` `  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example	"NAME:Contents", "MeasurementGeometry:=", ["LayerName:=", "TopLayer", "lw:=", 1.0, "sx:=", 0.0, "sy:=", 0.0, "ex:=", 10.0, "ey:=", 10.0], "text_style:=", ["name:=", "MyMeasure", "isPlot:=", True, "font:=", "Arial", "size:=", 12, "angle:=", 0, "weight:=", "Normal", "just:=", "Left", "mirror:=", False, "scales:=", True]])

CreateNportCircuitElements

Creates an HFSS Circuit Element from one or more ports.

UI Access	Right-click **Circuit Elements** > **Create Single Port Model...** or **Circuit Elements** > **Create Multi-Terminal Model...**
Parameters	Name: *<NPortArray>*; Type: Array; Description: Structured array. ["NAME:<NPortName>", "Definition:=", <string>, <AssignmentArray>]
Return Value	None.

Python Syntax	CreateNportCircuitElement (*<NPortArray>*)
Python Example	oModule.CreateNportCircuitElement(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:Nport1",
  "Definition:=", "Model",
  ["NAME:Assignments",
    ["NAME:Model",
    "Assign:=", "1"]
  ]
])
oModule.CreateNportCircuitElement(
["NAME:Nport2",
  "Definition:=", "Model2",
  ["NAME:Assignments",
    ["NAME:P01__NET179__T1",
    "Assign:=", "2"],
    ["NAME:P02__NET178__T1",
    "Assign:=", "3"],
    ["NAME:P03__NET178__T1",
    "Assign:=", "4"],
    ["NAME:P04__NET179__T1",
    "Assign:=", "5"],
  ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreateNetClass

*Use:*Create a new net class into a layout.

*Command:*CreateNetClass.

*Syntax:*CreateNetClass (<name> <description> <nets name list>)

*Return Value:*None

CreateObjectFromPolygon (Layout Editor)

Create a polygon on the specified layer and net.

UI Access	None.
Parameters	Name / Type / Description <*oPolygon*> / Object / A polygon object used to define the new layout polygon. <*Layer*> / String / The name of the layer on which the new polygon will be created. <*Net*> / String / The net to assign to the new polygon. Optional.
Return Value	New polygon name.

Python Syntax	oLayout.CreateObjectFromPolygon(<oPolygon>, <Layer>, <Net>)

Python Example	`oLayout.CreateObjectFromPolygon(`
	`  oPolygon,`
	`  "Layer1",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Net1")

CreatePin (Layout Editor)

Creates a pin.

UI Access	None.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Pin_Description*></td><td>Array</td><td>Array consisting of the following parameters in order: <*NAME:Contents*> <*Port*> <*Rotation*> <*Offset*> <*Padstack*> </td></tr><tr><td><*NAME:Contents*></td><td>String</td><td>Object descriptor header.</td></tr><tr><td><*Port*></td><td>Array</td><td>Port definition array (not used).</td></tr><tr><td><*Name*></td><td>String</td><td>Object name (unused).</td></tr><tr><td><*Rotation*></td><td>Array</td><td>Rotation angle of the pin.</td></tr><tr><td><*Value*></td><td>Double</td><td>Rotation angle value.</td></tr><tr><td><*Offset*></td><td>VPoint</td><td>Pin position.</td></tr><tr><td><*Padstack*></td><td>String</td><td>Padstack name.</td></tr></table>
Return Value	Pin name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	oEditor.CreatePin(<Pin_Description>)

Python Example	`oEditor.CreatePin(`
	`  [`
	`  "NAME:Contents",`
	`  ["NAME:Port", "Name:=", "MyObject"],`
	`  "Rotation:=", [90],`
	`  "Offset:=", [0.0, 0.0],`
	`  "Padstack:=", "MyPadstack"`
	`  ]`
	`)`

CreatePinGroup (Layout Editor)

Use: Create a pin group from a selection of pins.

Syntax: **CreatePinGroup(STRING *pin-group-name*, ["NAME:elements", STRING *pin*, STRING *pin*, …])**

CreatePinGroupPort (Layout Editor)

Use: Create a pin group port from two pin-groups; first is positive, second is negative.

Syntax: **CreatePinGroupPort(["Name:elements", STRING *pos-pin-group*, STRING *neg-pin-group*])**

CreatePinGroupPortsOnComponents (Layout Editor)

Creates pin group ports on a selected component.

UI Access	After selecting a component, right-click within the **Layout Editor** and select **Port** > **Create Ports on Component**.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Components>
<ComponentName></td></tr>
<tr><td><NAME:Components></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Components").</td></tr>
<tr><td><ComponentName></td><td>String</td><td>Names of the target components, separated by commas.</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:Nets>
<NetNames></td></tr>
<tr><td><NAME:Nets></td><td>String</td><td>Required first parameter which must have the specified value (i.e., "Nets").</td></tr>
<tr><td><NetNames></td><td>String</td><td>Names of the nets to place ports on, separated by commas.</td></tr>
<tr><td><ReferenceNet></td><td>String</td><td>Name of the net to use as a reference.</td></tr>
<tr><td><PinGroupPerNet></td><td>Boolean</td><td>True creates a different pin group for each net.
False creates a single pin group across all nets.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreatePinGroupPortsOnComponents([<NAME:Components>,<ComponentName>], [<NAME:Nets>, <NetNames>], <ReferenceNet>, <PinGroupPerNet>)

Python Example	`oEditor.CreatePinGroupPortsOnComponents(`
	`[`
	`  "NAME:Components",`
	`  U0`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   ],
   ["NAME:Nets",
   "net3",
   "net4"
   ], "net5", False
])
```

CreatePolygon (Layout Editor)

Creates a polygon primitive object.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.
	<LineGeometry>	Array	Array of geometric properties for the polygon, consisting of the following parameters in order: • *<LayerName>* • *<Lw>* • *<N>* • *<X0>* • *<Y0>* • *<X1>* • *<Y1>*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <*X2*> • <*Y2*> • <*X3*> • <*Y3*>
	<*LayerName*>	String	Name of the layer.
	<*Lw*>	Integer	Border line width.
	<*N*>	Integer	Number of polygon vertices.
	<*X0*>	Double	X-coordinate of vertex 0.
	<*Y0*>	Double	Y-coordinate of vertex 0.
	<*X1*>	Double	X-coordinate of vertex 1.
	<*Y1*>	Double	Y-coordinate of vertex 1.
	<*X2*>	Double	X-coordinate of vertex 2.
	<*Y2*>	Double	Y-coordinate of vertex 2.
	<*X3*>	Double	X-coordinate of vertex 3.
	<*Y3*>	Double	Y-coordinate of vertex 3.
Return Value	Object name (string)		

Python Syntax	CreatePolygon([<*NAME:Contents*>, [<*LineGeometry*>]])

Python Example	`oEditor.CreatePolygon(`
	`  [`
	`  "NAME:Contents",`
	`  "LineGeometry:=", [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "LayerName:=", "TopLayer",
        "Lw:=", 2,
        "n:=", 4,
        "x0:=", 10,
        "y0:=", 15,
        "x1:=", 25,
        "y1:=", 30,
        "x2:=", 40,
        "y2:=", 20,
        "x3:=", 15,
        "y3:=", 5
      ]
   ]
)
```

CreatePolygonVoid (Layout Editor)

Creates a polygon void and adds it to a specified parent primitive.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.
	<Owner>	String	Name of the parent primitive.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*Poly VoidGeometry*>	Array	Array of X,Y coordinate pairs defining the polygon void.
	<*X*>	Integer	X coordinate of a polygon vertex.
	<*Y*>	Integer	Y coordinate of a polygon vertex.
Return Value	Void object name.		

Python Syntax	CreatePolygonVoid([<*NAME:Contents*>, <*Owner*>, <*Poly VoidGeometry*>, [<*X*>, <*Y*>, ...]])

Python Example	`oEditor.CreatePolygonVoid(`
	`  [`
	`  "NAME:Contents",`
	`  "Owner:=", "TopLayer",`
	`  "Poly VoidGeometry:=", [`
	`  "X:=", "0mm",`
	`  "Y:=", "0mm",`
	`  "X:=", "10mm",`
	`  "Y:=", "0mm",`
	`  "X:=", "10mm",`
	`  "Y:=", "10mm",`
	`  "X:=", "0mm",`
	`  "Y:=", "10mm"`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]])

CreatePortInstancePorts (Layout Editor)

Use: Create port on port instances.

Command: Right-Click-Menu > Port > Create

Syntax: CreatePortInstancePorts (["NAME:elements", "element-name",…])

Return Value: None

Parameters: <element-name>

Type: string

// Name of a port instance on which ports will be created.

CreatePortsOnComponents (Layout Editor)

Creates port-on-port instances of selected components.

UI Access	Right click on component(s) to select **Port** > **Create Ports on Component**
Parameters	Name / Type / Description *<Name>* / String / "NAME:elements" *<element-name>* / String / Name of a component
Return Value	None. Ports are created on the unconnected port instances of the selected components.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreatePortsOnComponents(["NAME:elements", <element-name>,…])

Python Example	`oEditor.CreatePortsOnComponents(`
	`  [`
	`  "NAME:elements",`
	`  "poly101",`
	`  "rect100"`
	`  ])`

CreateRectangle (Layout Editor)

Creates a rectangle primitive object and adds it to the current layout.

UI Access	None		
Parameters	Name	Type	Description
	<NAME:Contents>	String	Object descriptor header.
	<RectGeometry>	Array	Array of geometric properties for the rectangle.
	<LayerName>	String	Name of the placement layer.
	<Lw>	Integer	Line width.
	<X>	Integer	Center X coordinate.
	<Y>	Integer	Center Y coordinate.
	<W>	Integer	Width (X-direction size).
	<H>	Integer	Height (Y-direction size).
	<Ang>	Integer	Rotation angle.
Return Value	Object name (string)		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateRectangle([<*NAME:Contents*>, <*RectGeometry*>, [<*LayerName*>, <*Lw*>, <*X*>, <*Y*>, <*W*>, <*H*>, <*Ang*>]])

Python Example	`oEditor.CreateRectangle(`
	`  [`
	`  "NAME:Contents",`
	`  "RectGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "Lw:=", 2,`
	`  "X:=", 10,`
	`  "Y:=", 15,`
	`  "W:=", 30,`
	`  "H:=", 20,`
	`  "Ang:=", 45`
	`  ]`
	`  ]`
	`)`

CreateRectangleVoid (Layout Editor)

Creates a rectangle void and adds it to a specified parent primitive.

UI Access	None		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<NAME:Contents>	String	Object descriptor header.
	<Owner>	String	Name of the parent primitive.
	<Rect VoidGeometry>	Array	Array defining the rectangle void geometry by two pairs of X,Y coordinates.
	<X1>	Integer	Start X coordinate.
	<Y1>	Integer	Start Y coordinate.
	<X2>	Integer	End X coordinate.
	<Y2>	Integer	End Y coordinate.
Return Value	Void object name.		

Python Syntax	CreateRectangleVoid([*<NAME:Contents>*, *<Owner>*, *<Rect VoidGeometry>*, [*<X1>*, *<Y1>*, *<X2>*, *<Y2>*]])

Python Example	`oEditor.CreateRectangleVoid(`
	`  [`
	`  "NAME:Contents",`
	`  "Owner:=", "TopLayer",`
	`  "Rect VoidGeometry:=", [`
	`  "X1:=", "0mm",`
	`  "Y1:=", "0mm",`
	`  "X2:=", "10mm",`
	`  "Y2:=", "5mm"`
	`  ]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreateText (Layout Editor)

Creates a text primitive object and adds it to the current layout.

UI Access	None
Parameters	See table below
Return Value	Object name (string)

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*TextGeometry*>	Array	Array of geometric and formatting properties for the text.
<*LayerName*>	String	Name of the placement layer.
<*X*>	Integer	Origin X coordinate.
<*Y*>	Integer	Origin Y coordinate.
<*Ang*>	Integer	Rotation angle.
<*IsPlot*>	Boolean	Whether the text is plotted.
<*Font*>	String	Font name.
<*Size*>	Integer	Font size.
<*Weight*>	Integer	Text weight.
<*Just*>	Integer	Text justification.
<*Mirror*>	Boolean	Whether the text is mirrored.
<*Text*>	String	Text content to display.

Python Syntax	CreateText([<*NAME:Contents*>, <*TextGeometry*>, [<*LayerName*>, <*X*>, <*Y*>, <*Ang*>, <*IsPlot*>, <*Font*>, <*Size*>, <*Weight*>, <*Just*>, <*Mirror*>, <*Text*>]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.CreateText(`
	`  [`
	`  "NAME:Contents",`
	`  "TextGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "X:=", 10,`
	`  "Y:=", 15,`
	`  "Ang:=", 0,`
	`  "IsPlot:=", true,`
	`  "Font:=", "Arial",`
	`  "Size:=", 12,`
	`  "Weight:=", 3,`
	`  "Just:=", 4,`
	`  "Mirror:=", false,`
	`  "Text:=", "Hello World"`
	`  ]`
	`  ]`
	`)`

CreateTrace (Layout Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Creates a trace with a manually specified path between pins, ports, or selected edges.

UI Access	From **Layout Editor**, navigate to **Draw** and select **Route** > **Manual**.

Parameters

Name	Type	Description
<Options>	Array	A structured array containing layer pairings for routing: <Layer1>, <Layer2>.
<Layer1>	String	Name of the starting layer.
<Layer2>	String	Name of the ending layer.
<Lines>	Array	-A structured array defining one or more trace segments: <Num>, <Line Name>, <Layer Name>, <Line Width>, <End Style>, <Join Style>, <Num Vertices>, <Units>, <X>, <Y>.
<Num>	Integer	Number of line entries.
<Line Name>	String	Name of the trace line.
<Layer Name>	String	Layer the trace is drawn on.
<Line Width>	String	Width of the trace, including units.
<End Style>	Integer	Cap style for line ends.
<Join Style>	Integer	Corner join style.
<Num Vertices>	Integer	Number of vertex points.
<Units>	String	Units for coordinates (e.g., "mm").
<X>, <Y>	Integer	Vertex coordinates.
<Vias>	Array	A structured array defining via instances and geometry: <Num>, <Via Name>, <Referenced Padstack>, <X>, <Y>, <Rotation>, <Override Hole>, <Hole Diameter>, <Highest Layer>, <Lowest Layer>.
<Num>	Integer	Number of via entries.
<Via Name>	String	Name of the via.
<Referenced Pad-stack>	String	Padstack definition name.
<X>, <Y>	Integer	Coordinates for via placement.
<Rotation>	String	Via orientation angle, including units (e.g., "180deg").

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Override Hole>	Boolean	If true, overrides the default hole diameter.
	<Hole Diameter>	String	Override diameter value, including units.
	<Highest Layer>	String	Top layer the via connects to.
	<Lowest Layer>	String	Bottom layer the via connects to.
	<Elements>	Array	A structured array listing ports or pins to connect: <Port Name>.
	<Port Name>	String	Name of the port or pin.
	<EdgeRefs>	Array	A structured array referencing primitive or via edges for trace attachment: <Primitive Name>, <Edge Number>, <Via Name>, <Layer ID>, <Start X>, <Start Y>, <End X>, <End Y>, <Arc Height>, <Radians>.
	<Primitive Name>	String	Name of the primitive containing the edge.
	<Edge Number>	Integer	Index of the edge on the primitive.
	<Via Name>	String	Name of the via used for arc reference.
	<Layer ID>	Integer	ID of the layer containing the arc.
	<Start X>, <Start Y>	Integer	Coordinates of the arc start point.
	<End X>, <End Y>	Integer	Coordinates of the arc end point.
	<Arc Height>	Integer	Height of the arc (0 for straight edge).
	<Radians>	Integer	Arc size in radians (0 for straight edge).
Return Value	None		

Python Syntax	CreateTrace([[<Options>], [<Lines>], [<Vias>], [<Elements>], [<EdgeRefs>]])

Python Example	`oEditor.CreateTrace([`
	`  ["NAME:Options", "Layer1:=", "Top", "Layer2:=", "Bottom"],`
	`  ["NAME:Lines", "Num:=", 1, "l0:=", ["Name:=", "Trace1", "LayerName:=", "Top",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Lw:=", "1mm", "EndStyle:=", 0, "JoinStyle:=", 1, "N:=", 2, "U:=", "mm", "X:=", 10, "Y:=", 20]], ["NAME:Vias", "Num:=", 1, "NAME:V1", "Name:=", "Via1", "ReferencedPadstack:=", "Pad1", "VPosition:=", ["X:=", 10, "Y:=", 20], "VRotation:=", ["180deg"], "Overrides Hole:=", True, "Hole Diameter:=", ["1mm"], "Highest Layer:=", "Top", "Lowest Layer:=", "Bottom"], ["NAME:Elements", "Port1"], ["NAME:EdgeRefs", "Edge:=", ["Et:=", "Pe", "Prim:=", "Prim1", "Edge:=", 3]]])

CreateVia (Layout Editor)

Use: Creates a new via.

Syntax: **CreateVia** <via_description>

Return Value: Returns the name of the created via

Parameters: <via_description>:

```
Array("NAME:Contents",
"name:=", <object_name>, // not used
"vposition:=", <vpoint>, // position
"rotation:=", double, // rotation in radians
"overrides hole:=", <bool>,// overrides or not padstack hole
"hole diameter:=", Array(<value>), // via diameter
"ReferencedPadstack:=", <quoted_string>, // padstack name
"highest_layer:=", <layer_name>,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"lowest_layer:=", <layer_name>)

CutOutSubDesign (Layout Editor)

Use: Cut out a subdesign.

Command: CutOutSubDesign

Syntax: oEditor.CutOutSubDesign Array(

"NAME:Params",

"Name:=", <"name">,

"EMDesign:=", <boolean>,

"SubDesign:=", <boolean>,

"Within:=", <boolean>,

"Without:=", <boolean>,

"AutoGenExtent":=<boolean>,

"Expansion":=<double>,

"RoundCorner":=<boolean>,

"Increments":=<integer>,

"ExtentSel:=", Array(<"extent-poly">, …),

Array("NAME:Nets", "net:=", Array(<"net-name">, <clip>), …))

In place of "ExtentSel:=", Array(<"extent-poly">, …) an explicit polygon can be used:

"Extent:=", Array("cl:=", true,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"pt:=", Array(U:=", <"units">, "x:=", <double>, "y:=" <double>, …))

Parameters: Name — name of the cutout design.

EMDesign — if true, create an EM design, otherwise create a Nexxim design.

Within — boolean; cut out the interior region.

Without — boolean; cut out the exterior region.

AutoGenExtent — boolean; when true, Circuit disregards both ExtentSel and Extent. Circuit will instead generate a polygonal shape around all nets which are not clipped.

Expansion — double; similar to the Auto Generate Extent dialog box. However, this is always a unitless fraction. By default it is set to .1.

RoundCorner — boolean; identical to the Auto Generate dialog box. By default this is set to true. In general, rounded corners are preferred when the cutout shape has acute or nea-acute angles

Increments — integer; this is from the Auto Generate Dialog, and by default is set to true. This can greatly increase the running time and a small increase can make a big difference. It is probably best to experiment with this parameter first, rather than set it arbitrarily.

ExtentSel— an array of extent polygon names.

Extent — alternative to ExtentSel; an explicit polygon defined by coordinates.

cl — must always be true; indicates that the polygon is closed.

U — coordinate units, e.g. "mm"

pt — array of coordinate values.

x — x coordinate value

y — y coordinate value

Nets — the net information.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

net — array of net information.

<"net-name">

"<no net trace>" — special value referring to traces not in a net.

"<no net fill>" — special value refereeing to fill polygons not in a net.

<design>:<net> — net within a particular design.

<net> — net within the active design.

<clip> — boolean true/false; if true, the net is clipped against the extent else the net is included but not clipped.

Return Value: None

Example: Using "extent_poly" as the selection extent:

```
oEditor.CutOutSubDesign Array("NAME:Params",
"Name:=", "EMDesign1_cutout",
"EMDesign:=", true,
"SubDesign:=", false,
"Within:=", true,
"Without:=", false,
"ExtentSel:=", Array("extent_poly", …),
Array("NAME:Nets",
"net:=", Array("<no net trace>", true),
"net:=", Array("<no net fill>", true),
"net:=", Array("EMDesign1:GND", true) … ))
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example: Example using an explicit polygon as the selection extent:

```
oEditor.CutOutSubDesign Array("NAME:Params",
"Name:=", "EMDesign1_cutout",
"EMDesign:=", true,
"SubDesign:=", false,
"Within:=", true,
"Without:=", false,
"Extent:=", Array(
"cl:=", true,
"pt:=", Array(
"U:=", "mm",
"x:=", 0,
"y:=" 0, ) ),
Array("NAME:Nets",
"net:=", Array("<no net trace>", true),
"net:=", Array("<no net fill>", true),
"net:=", Array("EMDesign1:GND", true) … ))
```

DeactivateOpen (Layout Editor)

Use: Causes deactivation of one or more components to *open circuit* state.

Syntax: **ActivateOpen** Array ("NAME:components",

<object_name>, // 1st component name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<object_name>, // 2nd component, if any

…) // etc

DeactivateShort (Layout Editor)

Use: Causes deactivation of one or more components to *short circuit* state.

Syntax: **ActivateShort** Array("NAME:components",

<object_name>, // 1st component name

<object_name>, // 2nd component, if any

…) // etc

Defeature Objects (Layout Editor)

Use: Remove irrelevant features from a primitive. Vertices that deviate less than the tolerance from a chord are removed; narrow concave regions less than the tolerance wide are also eliminated. The command can also be used to fix self-intersections (only the 'positive' areas are kept).

Command: DefeatureObjects

Syntax: DefeatureObjects Array("NAME:options", "tolerance:=", <value>, "fix:=", <fix>), Array("NAME:elements", <"primitive">, …)

Return Value: None

Parameters: <"value">

Type: double, e.g. 0.001

Description: tolerance value

<fix>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: boolean (true or false)

Description: if true, then self-intersections are fixed.

< primitive >

Type: text

Description: primitive name.

DelNetClass

*Use:*Delete a net class in a layout.

*Command:*DelNetClass

*Syntax:*DelNetClass (<names>)

*Return Value:*None

Delete (Layout Editor)

Use: Deletes one or more objects.

Syntax: **Delete** <object_name_list>// names of the objects to be deleted

`<object_name_list>:`

`Array ("NAME:elements", <object_name>, <object_name>, …)`

DeleteNets (Layout Editor)

Removes specified nets from the project.

UI Access	None.		
Parameters	Name	Type	Description
	<netName>	String	Name of unwanted net.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	DeleteNets ([<*netName*>, ...])

Python Example	`oEditor.DeleteNets(`
	`  [`
	`  "NET_1",`
	`  "NET_2",`
	`  "NET_3"`
	`  ])`

DeletePinGroup (Layout Editor)

Use: Deletes a pin group.

Syntax: **DeletePinGroup(STRING *pin-group-name*)**

Disconnect (Layout Editor)

Use: Removes the specified object(s) from their current electrical net(s) (if any).

Syntax: **Disconnect** Array ("NAME: *elements*",

`<object_name>, // 1st object`

`<object_name>, // 2nd object, if any`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

…) // etc

Duplicate (Layout Editor)

Use: Duplicates layout objects.

Command: The specified objects are duplicated by the given count with the specified offset.

Syntax: Duplicate Array("NAME:options", "count:=", <count data>), Array("NAME:elements", <element names>), Array(<x>, <y>)

Return Value: None

Parameters: <count data> is the number of sets of new objects to be generated (and does not include the original objects)

<element names> is one or more strings with the names of layout objects

<x> is the x value for the offset

<y> is the y value for the offset

DuplicateAcrossLyrs (Layout Editor)

Use: Duplicate selected objects (layout and footprint) to other layers.

Command: Draw > Duplicate > Across Layers

Syntax: DuplicateAcrossLyrs Array("NAME:elements", "element-name",…), Array("NAME:layers", "layer-name",…)

Return Value: None

Parameters: <*element-name*> // The name of the element to be duplicated.

<*layer-name*> // The name of the layer to duplicate elements to.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit (Layout Editor)

Use: Causes modification of one or more existing object(s).

Syntax: **Edit** <Array("NAME:items".....)>,

Parameters: <Array("NAME:items"
<edit_object_info>, // one object info

<edit_object_info>, // another one

…) // etc

<edit_object_info>:

Array("NAME:item",

"name:=", <object_name>, // name of the object

<object_description>) // 'new' object state, type should be consistent with <object_name>:

<object_description>:

<circle_description> |

<rectangle_description> |

<line_description> |

<polyon_description> |

<text_description> |

<circle_void_description> |

<rectangle_void_description> |

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<line_void_description> |
<polyon_void_description> |
<component_description> |
<pin_description> |
<via_description>
```

Edit3DComponentDefinition

Edits definitions of a specified 3D component.

<table>
<tr><td>UI Access</td><td colspan="3">Layout > 3D Component Definitions. Select component and click Edit Definition.</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
<code>["NAME:3D Components",</code>
<code>"Definitions:=" , [<string component name>],</code>
<code>"IsLocal:=" , <boolean True for local definition; False to save to file>,</code>
<code>"EditFilePath:=", <string path of new component file location>]</code>
If saving locally or leaving the file path unaltered, pass empty string for <code>EditFilePath</code>.</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
<tr><td>Python Syntax</td><td colspan="3">Edit3DComponentDefinition (<Parameters>)</td></tr>
<tr><td>Python Example</td><td colspan="3"><code>oEditor.Update3DComponentDefinitions(</code></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:3D Components",
	  "Definitions:=" , ["My_Component"],
	  "IsLocal:=" , True,
	  "EditFilePath:=", ""
	)

EraseMeasurements (Layout Editor)

Use: Causes erasing of ALL measurements currently present.

Command: None.

Syntax: EraseMeasurements

Return Value: None.

Parameters: None.

Expand (Layout Editor)

Scales an existing circle, square, or polygon. To expand an object while preserving ports, use ExpandWithPorts.

UI Access	**Draw** > **Expand**.		
Parameters	Name	Type	Description
	<expansion>	String	Expansion amount, including unit. Value can be negative to shrink an object.
	<join>	String	Corner style. One of: • "ROUND" • "MITER" • "CORNER"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<replace>	Boolean	True to replace original object; False to create new object.

	<items>	Array	Structured array containing object to expand:
			`["NAME:elements",`
			`"<Name of Object to Expand>"]`

Return Value	None.		

Python Syntax	Expand (*<expansion>*, *<join>*, *<replace>*, *<items>*)
Python Example	`oEditor.Expand("2mm", "ROUND", True, ["NAME:elements", "circle_0"])`

ExpandWithPorts (Layout Editor)

Scales an existing circle, square, or polygon and preserves ports. To expand an object without preserving ports, use Expand.

UI Access	**Draw** > **Expand**.		
Parameters	Name	Type	Description
	<expansion>	String	Expansion amount, including unit. Value can be negative to shrink an object.
	<join>	String	Corner style. One of: • "ROUND" • "MITER" • "CORNER"
	<items>	Array	Structured array containing object to expand (i.e., ["NAME:elements, "<name of object to expand>"]).
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExpandWithPorts (<*expansion*>, <*join*>, <*items*>)
Python Example	`oEditor.ExpandWithPorts("2mm", "ROUND", ["NAME:elements", "circle_0"])`

ExportDXF (Layout Editor)

Use: Export DXF

Command: File > Export > AutoCAD

Syntax: ExportDXF

Array("NAME:options", "FileName:=", <"filename">, "AcadVersion:=,<version>, "ScaleFactor:=", <scale>),

Array("NAME:layers", <"layer">, …)

Return Value: None

Parameters:

<"filename">

Type: text

Description: file path for the export file

<version>

Type: integer

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Description: version of AutoCAD. Acceptable values are:

- **0** corresponding to AutoCAD version 13
- **1** corresponding to AutoCAD version 14
- **2** corresponding to AutoCAD 2000-2003
- **3** corresponding to AutoCAD 2004-2006
- **4** corresponding to AutoCAD 2007-2009
- **5** corresponding to AutoCAD 2010-2012

<scale>

Type: double

Description: the scaling of the values written out, e.g. if the data is written out in "mm", the scaling factor will be 1000.

<"layer">

Type: text

Description: names of the layers to be exported.

ExportGDSII (Layout Editor)

Exports results to GDSII.

<table>
<tr><td>UI Access</td><td colspan="3">File > Export > GDSII.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><OptionParameters></td><td>Array</td><td>Structured array.
["NAME:options",
"FileName:=", <string, GDSII export file name.>,
"NumVertices:=", <integer, maximum num-</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ber of vertices allowed in a polygon (specify zero if unlimited).>, "ArcTol:=", <double, arc tolerance (in meters); the value used to discretize arcs. Smaller the value the greater the number of edges exported. For 0.1mm use 0.0001.>, "LayerMap:=", ["entry:=", ["layer:=", <string, name of a layer to export.>, "id:=", <integer, GDSII layer ID to assign the exported layer.>, "include:=", <boolean, if true, the layer is exported.>]]]
Return Value	None.		

Python Syntax	ExportGDSII(<*OptionParameters*>)
Python Example	oEditor.ExportGDSII(["NAME:options",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"FileName:=", "C:/Projects/ design.gds",
"NumVertices:=", 0,
"ArcTol:=", 2E-006,
"LayerMap:=", [
        "entry:=",[
                "layer:=", "Bottom",
                "id:=", 2,
                "include:=", true],
        "entry:=", [
                "layer:=", "Top",
                "id:=", 1,
                "include:=", true]
        ]
]
```

ExportGerber (Layout Editor)

Use: Export the current design to Gerber files.

Command: File > Export > Gerber

Syntax: ExportGerber

Array("NAME:options",

"FileName:=", <"file name">,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Units:=", <"units">,

"IntPlace:=", <integer places>,

"DecimalPlace:=", <decimal places>,

"Suppress:=", <"suppress">,

Array("NAME:GerberPages",

"<page>:=", Array(<"layer">, …), …))

Return Value: None

Parameters: <"file name">

Type: text

Description: prefix for the exported Gerber files; the actual files will be named <file name>_<page>.ger

<"units">

Type: text

Description: unit code, e.g. "in", "mm", etc.

<integer places>

Type: integer

Description: number of integer places to use when formatting the numerical output.

<decimal places>

Type: integer

Description: number of decimal places to use when formatting the numerical output.

<"suppress">

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: text

Description: either "Leading Zeros", or "Trailing Zeros".

<page>

Type: integer

Description: Gerber page number

<"layer">

Type: text

Description: Layers to be exported to the specified Gerber page.

ExportNCDrill (Layout Editor)

Use: Export the vias to an NC Drill file

Command: File > Export > NCDrill

Syntax: ExportNCDrill

```
Array("NAME:options",
"FileName:=", <"file name">,
"Units:=", <"units">,
"IntPlace:=", <integer places>,
"DecimalPlace:=", <decimal places>,
"SuppressLeadingZero:=", <suppress leading zeros>,
Array("NAME:NCDFileOptsVec",
Array("NAME:NCDrillOptions",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Top:=", <"layer">,

"Bottom:=", <"layer">,

"Create:=", <create>), …))

Return Value: None

Parameters: <"file name">

Type: text

Description: prefix for the exported Gerber files; the actual files will be named <file name>_<page>.ger

<"units">

Type: text

Description: unit code, e.g. "in", "mm", etc.

<integer places>

Type: integer

Description: number of integer places to use when formatting the numerical output.

<decimal places>

Type: integer

Description: number of decimal places to use when formatting the numerical output.

<suppress leading zeros>

Type: boolean

Description: if true, then suppress leading zeros in the output file.

<layer>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: text

Description: Start/end layer names; each layer range is written to a separate NC drill file.

`<create>`

Type: boolean

Description: if true, create the corresponding NC drill file.

ExportToHFSS (Layout Editor)

Use: Exports an HFSS 3D Layout design to a new project with an equivalent HFSS 3D design. This command is only valid on HFSS Setup types.

Syntax: ExportToHfss(<setupName>, <pathToExportFile>)

ExportToQ3D (Layout Editor)

Use: Exports an HFSS 3D Layout design to a new project containing an equivalent Q3D design. This command is only valid on HFSS Setup types.

Syntax: ExportToQ3d(<setupName>, <pathToExportFile>)

FilterObjectList (Layout Editor)

Filters a list of object names using a key/value pair. See: FindObjects.

UI Access	N/A.		
Parameters	Name	Type	Description
	<key>	String	One of: • **"Name"** to search by object name. • **"Type"** to search by object type.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			Valid <value> strings for this type include: 'pin', 'via', 'rect', 'arc', 'line', 'poly', 'plg', 'circle void', 'line void', 'rect void', 'poly void', 'plg void', 'text', 'cell', 'Measurement', 'Port', 'Port Instance', 'Port Instance Port', 'Edge Port', 'component', 'CS', 'S3D', 'ViaGroup' • **"Layer"** to search by layer. An asterisk ('*') in the <value> string matches all layers. Vias and pins match using their defined layer range. 'Multi' matches (non via/pin) multi-layer objects. • **"Net"** to search by net assignment.
	<*value*>	String	Specific query. Queries use a basic string match. For instance, an asterisk ('*') matches anything. All matching is case insensitive.
Return Value	Filtered list of object names.		

Python Syntax	FilterObjectList (<*key*>, <*value*>)
Python Example	`RFIN_pins = oLayout.FilterObjectList('Type', 'Pin', oLayout.FindObjects('Net','RFIN'))`

FindObjectsByPoint (Layout Editor)

Use: Get a list of all objects intersected by the specified Point object on a given layer.

Syntax: oLayout.FindObjectsByPoint(<oPoint>, <Layer>)

Return Value: Object list.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

FindObjectsByPolygon (Layout Editor)

Use: Get a list of all objects intersecting the specified Polygon object on the given layer.

Syntax: oLayout.FindObjectsByPolygon(<oPolygopn>, <Layer>)

Return Value: Object list.

FlipHorizontal (Layout Editor)

Use: Causes horizontal flipping of one or more specified objects (mirroring about a vertical axis given its X coordinate).

Syntax: **FlipHorizontal** <object_name_list>,//objects to be flipped <position_or_center> // (X of flip_axis / not used)

FlipVertical (Layout Editor)

Use: Causes vertical flipping of one or more specified objects (mirroring about a horizontal axis given its Y coordinate).

Syntax: **FlipVertical** <object_name_list>, //objects to be flipped <position_or_center> // not used, flip axis

GeometryCheckAndAutofix

Use: Runs Geometry Check and optionally applies autofixes.

Command: **HFSS 3D Layout > Geometry Check**

Syntax: GeometryCheckAndAutofix <ChecksArray>,

minimum_area_meters_squared,

<FixesArray>

Return Value: None

Parameters: **<ChecksArray> - Array("NAME:checks", <check 1>, <check 2>, …, <check n>)**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Specify the checks that should be included. Specifying fewer checks may speed up execution but may also result in less problems reported in the message manager (or the logfile) and consequently less problems that can be fixed by autofixes.

The following are valid checks that can be specified:

- "Self-Intersecting Polygons"
- "Disjoint Nets (Floating Nodes)"
- "DC-Short Errors"
- "Identical/Overlapping Vias"
- "Misaligments"

There may be no checks, all 5 of the checks, or anything in between. The order that checks are specified in is not relevant.

minimum_area_meters_squared

Specify a decimal value for the minimum area (e.g. .000015) optionally in scientific notation (e.g. 2E-006). Cutouts smaller than this minimum area may be ignored during validation check.

<FixesArray> - Array("NAME:fixes", <fix 1>, <fix 2>, …, <fix n>)

Specify the autofixes that should be applied if they are found by a check.

The following are valid fixes that can be specified:

- "Self-Intersecting Polygons"
- "Disjoint Nets",
- "Identical/Overlapping Vias"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- "Traces-Inside-Traces Errors"
- "Misalignments (Planes/Traces/Vias)"

There may be no fixes specified, all 5 fixes specified, or anything in between. The order that fixes are specified in is not relevant.

GetAllLayerNames (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetAllLayerNames

Return Value: Array of strings which are the names of all layers in the layout, blackbox, or footprint.

GetBBox (Layout Editor)

Use: Get a Polygon Object defining the bounding box for the specified object name, or None if the object does not exist.

Syntax: oLayout.GetBBox(<Name>)

Return Value: Polygon object.

GetCSObjects (Layout Editor)

Use: Given a coordinate system name, returns a list of the objects inside. Coordinate system names can be found using FindObjects ('Type', 'CS'). See Find Objects.

Syntax: oLayout.GetCSObjects(<CS_name>)

Return Value: List of objects in the coordinate system.

GetCompInstanceFromRefDes (Layout Editor)

Use: Informational.

Command: None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: GetCompInstanceFromRefDes (<object_name>)

Return Value: Returns IDispatch for CompInstance.

Parameters: <object_name> // string is refDes

GetComponentInfo (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetComponentInfo<compID>

Return Value: array of strings, as follows:

"ComponentName=*string"*

"PlacementLayer=*string"*

"LocationX=*number"*

"LocationY=*number"*

"BBoxLLx=*number"*

"BBoxLLy=*number"*

"BBoxURx=*number"*

"BBoxURy=*number"*

"Angle=*number" (in degrees)*

"Flip=true *or* false"

"Scale=*number"*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: The LayoutComp's ID

GetComponentPinInfo (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetComponentPinInfo<compID, pinName>

Return Value: retval = array of strings, as follows:

"X=val"

"Y=val"

"Angle=val"

"Flip=true/false"

"WireID=string"

Parameters: compID = The LayoutComp's ID

pinName = The name of the pin

GetComponentPins (Layout Editor)

Use: Informational.

Command: None.

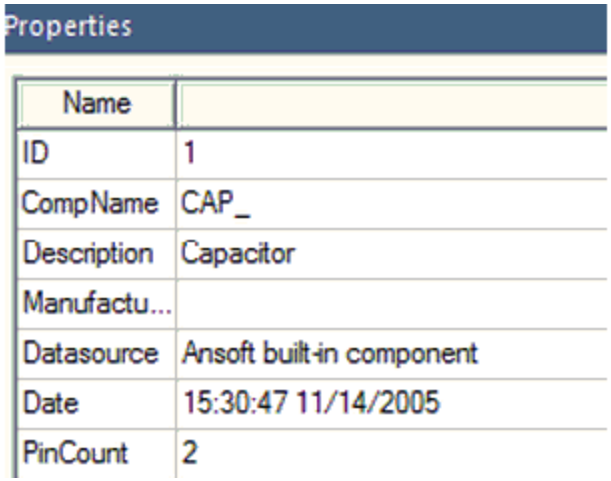
Syntax: GetComponentPins<compID>

Return Value: array of strings, which are the names of all the component's pins

Parameters: The LayoutComp's ID
CompInst@<ComponentName>;<CompInstID>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note: For the documented example, the capacitor component name is CAP_ and its ID is 1. If you can select the item of interest in the schematic, the **Properties** window gets updated with the corresponding component name, its ID, and other details as shown in the following figure.



GetEditorName (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetEditorName()

Return Value: Returns the name of the editor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetMaterialList (Layout Editor)

Use: Get the names of all the materials for a layout.

Command: None.

Syntax: GetMaterialList

Return Value: Array of strings.

Parameters: None.

```
materialNames = oEditor.GetMaterialList
```

GetNetClassNets

*Use:*Gets the list of nets in a net class.

*Command:*GetNetClassNets

*Syntax:*GetNetClassNets (<name>)

*Return Value:*None

GetNetClasses

*Use:*Gets all the net classes in a layout.

*Command:*GetNetClasses

*Syntax:*GetNetClasses ()

*Return Value:*None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetNetConnections (Layout Editor)

Use: Informational.

Command: None.

Syntax: GetNetConnections(<netName>)

Return Value: Array of strings containing the connections for the net identified by the netName argument. The strings in the array are in one of four formats:

"ComponentPin compID pinname x y EdgePort"

"ComponentPin compID pinname x y PinPadstack: padstackName"

"InterfacePort portname portID x y Padstack: padstackName"

"EdgePort portname portID EdgeInfo: Primitive id, edge index[Primitive id, edge index]"

where *compID* is the component instance identifier, *pinname* is the name of the connected pin, *x* and *y* are the connection point, *padstackName* is the name of the padstack, *portname* is the name of the port, *portID* is the identifier for the interface port, and *id* and *index* identify edges involved in an edgeport.

Parameters: <netName>

Type: String

The name of the net.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPolygon (Layout Editor)

Use: Get a Polygon Object for the specified object name, or None if the object does not exist. Returned polygon is as rendered, in SI units

Syntax: oLayout.GetPolygon(<Name>)

Return Value: Polygon object.

GetPolygonDef (Layout Editor)

Use: Get a Polygon Object for the specified object name, or None if the object does not exist. Returned polygon is as rendered, in SI units. For trace types, this corresponds to the center line.

Syntax: oLayout.GetPolygonDef(<Name>)

Return Value: Polygon object.

GetPortInfo (Layout Editor)

Use: Request information on a port or pin.

Command: None.

Syntax: GetPortInfo<"name">

Return Value: an array of text as follows:

For pins and ports that are not edge ports:

`Name=<name>`

Type=Pin, Padstack: <padstack definition name>

X=<X coordinate>

Y=<Y coordinate>

ConnectionPoints=<connection description>; <connection description>,...

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

NetName=<net name>

For edge ports:

Name=<name>

Type=EdgePort

X=<X coordinate>

Y=<Y coordinate>

ConnectionPoints=<connection description>; <connection description>,...

NetName=<net name>

<name>

Text containing the name of the pin or port

<padstack definition name>

Text containing the name of the associated padstack definition

<X coordinate>

Double indicating the X location of the pin or port location

<Y coordinate>

Double indicating the Y location of the pin or port location

<connection description>

< X coordinate> <Y coordinate> Dir:<direction> Layer: <layer name>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<direction>

Either NONE or a double giving the angle in degrees for the connection direction

<layer name>

Name of the layer for the connection point being described.

Example returned values:

Name=Pin1

Type=Pin, Padstack: Padstack

X=0.000744

Y=0.015537

ConnectionPoints= 0.000744 0.015537 Dir:NONE Layer: Top; 0.000744 0.015537 Dir:NONE Layer: Ground; 0.000744 0.015537 Dir:NONE Layer: Bottom

NetName=Pin1

Name=Port1

Type=EdgePort

X=-0.008120

Y=0.004264

ConnectionPoints= -0.008120 0.004264 Dir:90.000000 Layer: Bottom

NetName=Port1

Parameters: <"name">

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Text that contains the name of the port or pin for which information is being requested.

GetProperties (Layout Editor)

Use: Gets a list of all the properties belonging to a specific **PropServer** and **PropTab**. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetProperties(<PropTab>, <PropServer>)

Return Value: Variant array of strings – the names of the properties belonging to the prop server.

GetPropertyValue (Layout Editor)

Use: Gets the value of a single property. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetPropertyValue(<PropTab>, <PropServer>, <PropName>)

Return Value: String representing the property value.

GetSelections [Design]

This script serves no function at the Design level. See: GetSelections (Layout Editor), GetSelections (Model Editor), or Get Selections (Schematic Editor).

Group (Layout Editor)

Use: Groups a set of primitives into a coordinate system.

Command: None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: oEditor.Group <CS name>, Array("NAME:elements", <element1>, <element2>, …)

Return Value: None.

Heal (Layout Editor)

Heals imprecise geometry.

<table>
<tr><td>UI Access</td><td>From the Layout ribbon, click Healing.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><HealArray></td><td>Array</td><td>Structured array containing these parameters in order:
<NAME:category>
<Type>
<OptionalArgs>
<Tol></td></tr>
<tr><td><<i>NAME:category</i>></td><td>String</td><td>Required first parameter which identifies the category of healing (i.e., "Snap", "Feature", or "<i>Repair</i>", such as <NAME:Snap>.</td></tr>
<tr><td><<i>Type</i>></td><td>String</td><td>Dependent on <NAME:category>. Identifies the type of healing, per category:
Snap — "Point", "Arc", or "Grid"
Feature — "Voids" or "FloatingBodies"
Repair — "Colinear" or "SelfIntersecting"</td></tr>
<tr><td><OptionalArgs></td><td>String</td><td>Optional parameters (i.e., "Antipads", "ModifyArcs", and/or "InsertPointsInEdges". Multiple <OptionalArgs> can be entered, depending on <NAME:category> and <Type>.</td></tr>
<tr><td><Antipads></td><td>Boolean</td><td>If <NAME:Feature> and <Type>="Voids", then this optional argument is available:

True — voids are converted to antipads
False — voids are not converted to antipads</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<ModifyArcs>	Boolean	If <NAME:Snap> and <Point>="Voids", then this optional argument is available: **0** — arcs do not snap to close primitive edges **1** — arcs snap to close primitive edges
	<InsertPointsInEdges>	Boolean	If <NAME:Snap> and <Point>="Voids", then this optional argument is available: **0** — new points will not appear when healed primitives snap together **1** — new points will appear at midpoints between existing edges and at the new edges created when healed primitives snap together
	<*Tol*>	Integer	Primitives at or within this tolerance are healed.
Return Value	None		

Python Syntax	oEditor.Heal([<NAME:<category>, <Type>, <OptionalArgs>, <Tol>])

Python Example	`oProject = oDesktop.SetActiveProject("GeometryHealing")`
	`oDesign = oProject.SetActiveDesign("TestDesign")`
	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.Heal(`
	`  [`
	`  "NAME:Snap",`
	`  "Type:=" , "Point",`
	`  "ModifyArcs:=" , 1,`
	`  "InsertPointsInEdges:=" , 1,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Tol:=" , "2mm", "Tol:=" , "2mm"])

HighlightNet (Layout Editor)

Highlights or removes the highlight from all elements in a net.

UI Access	None.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*requiredPara-meterKeyword*></td><td>String</td><td>Required first parameter which must have the value "Name:Args".</td></tr><tr><td><*requiredNetKeyword*></td><td>String</td><td>Required second parameter which must have the value "Name:=". Must be the first parameter for each net.</td></tr><tr><td><*netName*></td><td>String</td><td>Name of the net.</td></tr><tr><td><*requiredFlagKeyword*></td><td>String</td><td>Required fourth parameter which must have the value "Hi:=". Must be the third parameter for each net.</td></tr><tr><td><*highlightFlag*></td><td>Boolean</td><td>True if the net should be highlighted. False if it should not.</td></tr></table>
Return Value	None.

Python Syntax	HighlightNet (["Name:Args", "Name:=", " <*netName*>", "Hi:=", <*highlightFlag*>, ...])
Python Example	oEditor.HighlightNet(["NAME:Args",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Name:=", "net_0", "Hi:=", True])

ImportDXF [Modeler]

Imports a DXF model file.

UI Access	**Modeler** > **Import**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:options", "FileName:=" , <string>, "Scale:=" , <float>, "AutoDetectClosed:=" , <boolean>, "SelfStitch:=" , <boolean>, "DefeatureGeometry:=" , <boolean>, "DefeatureDistance:=" , <integer>, "RoundCoordinates:=" , <boolean>, "RoundNumDigits:=" , <integer>, "WritePolyWithWidthAsFilledPoly:=", <boolean>, "ImportMethod:=" , <integer>, "2DSheetBodies:=" , <boolean>, <layersArray>]
	<layersArray>	Array	Structured array.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:LayerInfo", <layer>, <layer>, <layer>,...]

	<layer>	Array	Structured array.
			["NAME:<layerName>",
			  "source:=" , <string>,
			  "display_source:=" , <string>,
			  "import:=" , <boolean>,
			  "dest:=" , <string>,
			  "dest_selected:=" , <boolean>,
			  "layer_type:=" , <string>]

Return Value	None.		

Python Syntax	ImportDXF(<*Parameters*>)

Python Example	oEditor.ImportDXF(
	  ["NAME:options",
	  "FileName:=", "C:/Users/jdoe/Desktop/export.dxf",
	  "Scale:=" , 0.001,
	  "AutoDetectClosed:=" , True,
	  "SelfStitch:=" , True,
	  "DefeatureGeometry:=" , False,
	  "DefeatureDistance:=" , 0,
	  "RoundCoordinates:=" , False,
	  "RoundNumDigits:=" , 4,
	  "WritePolyWithWidthAsFilledPoly:=", False,
	  "ImportMethod:=" , 1,
	  "2DSheetBodies:=" , False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ["NAME:LayerInfo",
    ["NAME:0",
    "source:="              , "0",
    "display_source:="      , "0",
    "import:="              , True,
    "dest:="                , "0",
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ],
    ["NAME:LAYER_1",
    "source:="              , "LAYER_1",
    "display_source:="      , "LAYER_1",
    "import:="              , True,
    "dest:="                , "LAYER_1",
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ],
    ["NAME:LAYER_2",
    "source:="              , "LAYER_2",
    "display_source:="      , "LAYER_2",
    "import:="              , True,
    "dest:="                , "LAYER_2",
    "dest_selected:="       , False,
    "layer_type:="          , "signal"
    ]
  ]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportGDSII

Imports a GDSII file into a new project.

UI Access	**File** > **Import** > **GDSII**.
Parameters	

Name	Type	Description
<*gdsiiFileName*>	String	Full path of GDSII file.
<*outputPathName*>	String	Full path of EDB file to create during import.
<*controlFileName*>	String	Optional. Full path of XML control file. Full path of “technology” (corresponding to `-t=technologyfile` argument in anstranslator). Pass empty string if none.
<*mapFileName*>	String	If technology file was used in place of the control file. Full path to either gdsii mapping file (corresponding to `-g=gdsMappingFile` argument in anstranslator) or XML control file. Otherwise, pass empty string.

Return Value	None.

Python Syntax	ImportGDSII(<*gdsiiFileName*>, <*outputPathName*>, <*controlFileName*>, <*mapFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportGDSII('C:/Files/test.gds', 'C:/Files/test.aedb',`
	`'C:/Files/test.ircx', 'C:/Files/test.xml')`

Intersect (Layout Editor)

Use: Causes Boolean intersecting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: **Intersect** Array ("NAME: *primitives*",

<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any

…) // etc

ModifyNetClass

*Use:*Modify an existing net class in a layout.

*Command:*ModifyNetClass

*Syntax:*ModifyNetClass (<name> <new name> <new description> <new nets name list>)

*Return Value:*None

Move (Layout Editor)

Use: Translate this Polygon by the vector specified in the provided Point. Return this Polygon.

Syntax: oPolygon.Move(<oPoint>)

Return Value: This Polygon object.

PageSetup (Layout Editor)

Use: Specifies page setup for printing.

Command: **File>Page Setup**

Syntax: PageSetup <ArgArray>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None.

Parameters: `<Margins>: Page margins in implicit units of inches.`

<Border>: Integer value indicating to draw border (1) or not to draw (0).

<DesignVars>: Integer value indicating to draw design vars (1) or not to draw (0).

Point (Layout Editor)

*Use:*Create and return a Point Object.

*Syntax:*oLayout.Point()

*Return Value:*Point object.

Polygon (Layout Editor)

Use: Create and return a Polygon Object.

Syntax: oLayout.Polygon()

Return Value: Polygon object.

PositionRelative (Layout Editor)

Use: Position a coordinate system (CS), including 3D structures and components, relative to another object.

Command: Draw > Position Relative

Syntax: There are two forms available:

PositionRelative

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Array("NAME:Contents",
"RelPos:=", Array(<edge description>)),
Array("NAME:elements", <coordinate system name>, < coordinate system name >, …)

PositionRelative Array("NAME:Contents",
"RelPos:=", Array("t:=", "pr", "from:=", <edge port or pin name>)),
Array("NAME:elements", <coordinate system name>, <coordinate system name>, …)

Return Value: None.

Parameters:

Array("NAME:elements", "CS_23")

<edge description> for primitive edges

"t:=", "pe", "from:=", <"prim">, "pos:=", <edge position>, "et:=", "pe", "prim:=", <"prim">, "edge:=", <edge#>

<"prim">: text that is the primitive name

<edge position>
double between 0 and 1, inclusive

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

0 indicates the start of the edge

1 indicates the end of the edge

values in between are positions along the edge

<edge#>: integer that is the edge number on the primitive

<edge description> for via edges

"t:=", "pe", "from:=", <"via">, "pos:=", <edge position>, "et:=", "pse",
"sel:=", <"via">, "layer:=", <layer id>, "sx:=", <start X location>,
"sy:=", <start Y location>, "ex:=", <end X location>, "ey:=",
<end Y location>, "h:=", <arc height>, "rad:=", <radians>

<"via">: text that is the name of the via to use

<layer id>: an integer that is the id of the layer of the pad of the via to use

<start X location>, <start Y Location>:
  doubles that are the X, Y location of the start point of the edge arc

<end X location>, <end Y Location>:
  doubles that are the X, Y location of the end point of the edge arc

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

<arc height>: double giving the height of the edge arc (0 for a straight edge)

<radians>: double giving the arc size in radians (0 for a straight edge)

< coordinate system name >

Text that contains the name of the coordinate system

<edge port or pin name>

Text that contains the name of the edge port or via

PushExcitations

Allows access to computed excitations for transient and linear frequency solutions. The script command can be accessed from three locations.

<table>
<tr><td>UI Access</td><td>

Layout Editor
Schematic Editor
Select a Nexxim solution in a 3D Layout design, right click, and choose Push Excitations.

</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Reference Designation></td><td>String</td><td>"<refdes>"

This argument is empty when excitations are pushed from a Nexxim solution.</td></tr>
<tr><td><PushExcitations Parameters></td><td>Array</td><td>This parameter changes depending on whether the excitations comes from a transient or linear frequency solution. The keyword "<code>transient:=</code>" indicates to AEDT which solution generated the</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			excitations. If "transient:=" is present, "CalcThevenin =" and its value are ignored. For a linear frequency solution, use this array: ["NAME:options", "CalcThevenin =" , <true|false>, "Sol:=", "<solution name>"] If CalcThevenin is true, Thevenin's equivalent is calculated. Parameters for the linear frequency solution do not include a Freq argument, so all frequencies from the solution are used. For a transient solution, use: ["NAME:options", "transient:=", ["start:=", <start time>, "stop:=", <stop time>, "maxHarmonics:=", <max harmonics>, "winType:=", <window>, ["widthPct:=", <width percentage>,] ["kaiser:=", <Kaiser value>,] ["correctCoherentGain:=", true]], "Sol:=", "<solution name>"] winType can have the following values: • Rectangular

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Bartlett • Blackman • Hamming • Hanning • Kaiser • Welch • Weber • Lanzcos
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

RemoveLayer (Layout Editor)

Use: Removes a layer or stackup layer.

Command: Remove Layer from a layout or footprint definition

Syntax: RemoveLayer (<LayerName>)

Return Value: None.

Parameters: `<LayerName>`

Type: <String>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note As with other Layout scripting interface commands that modifiy the layout, this command is not intended for use within scripts that define footprints. The command behavior from within such a script is undefined and may be unexpected. Use the LayoutHost scripting interface commands within scripts that define footprints. Users can, however, execute the **RemoveLayer** command from both general and **IC** modes, although the command functions differently in each environment. Refer to RemoveLayer (IC Editor).

RemovePortsFromAllNets (Layout Editor)

Removes ports from all the pins in all the nets.

UI Access	Layout tab under Nets right-click and click **Remove Ports**.
Parameters	Name | Type | Description None | |
Return Value	None

Python Syntax	RemovePortsFromAllNets
Python Example	`oEditor.RemovePortsFromAllNets()`

RemovePortsFromNet (Layout Editor)

Removes ports from all the pins on the designated nets.

UI Access	In the **Layout** window, right click the net and click **Port** > **Remove Ports from Net** or on the **Nets** tab of the **Nets** window, highlight and right click one or more nets and choose **Remove Ports**.
Parameters	Name | Type | Description <*requiredKeyword*> | String | Required parameter that must proceed all element names with the value "NAME:Nets".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*netName*>	String	Name of a net. Ports are removed from the port instances of the selected nets.
Return Value	None.		

Python Syntax	RemovePortsFromNet(["NAME:Nets", <*netName*>, ...])

Python Example	`oEditor.RemovePortsFromNet(`
	`  [`
	`  "NAME:Nets",`
	`  "NET_1",`
	`  "NET_2",`
	`  "NET_3"`
	`  ])`

RemovePortsOnComponents (Layout Editor)

Remove ports on port instances of selected components. Multiple nets can be listed.

UI Access	Right click and choose **Port** > **Remove Ports From Component**.		
Parameters	Name	Type	Description
	<*requiredKeyword*>	String	Required parameter that must proceed all element names with the value "NAME:elements".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*elementName*>	String	Name of a component. Ports are removed from the port instances of the selected components.
Return Value	None.		

Python Syntax	RemovePortsOnComponents (["NAME:elements", <*elementName*>,...])

Python Example	`oEditor.RemovePortsOnComponents(`
	`  [`
	`  "NAME:elements",`
	`  "0"`
	`  ])`

Rotate (Layout Editor)

Use: Rotate this Polygon counter clockwise (CCW) about a center specified by the provided Point. Angle is in radians. Return this Polygon.

Syntax: oPolygon.Rotate(<AngRad>, <oPoint>)

Return Value: This Polygon object.

Example: `oPolygon = oPolygon.Rotate(pi/2, oLayout.Point)`

SelectAll (Layout Editor)

Use: Select all elements in the layout editor.

Command: None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: SelectAll()

Parameters: None

Example: Dim removedDefs

```
removedDefs = oDefinitionEditor.SelectAll()
```

SetCS (Layout Editor)

Use: Activate a coordinate system (CS).

Command: None.

Syntax: oEditor.SetCS <CS name or blank>

Return Value: None.

Example: `oEditor.SetCS "CS_7" // activate CS_7`

```
oEditor.SetCS "" // activate the 'global CS', i.e. no CS is active
```

SetLayerMapping (Layout Editor)

Use: Set layer mapping.

Command: None.

Syntax: SetLayerMapping <component name>, <design layer>, <footprint layer>

Return Value: None.

Parameters: <component name> is a string that specifies the name of the component

<design layer> is a string that specifies the name of the design layer

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<footprint layer> is a string that specifies the name of the footprint layer

Example: `oEditor.SetLayerMapping "2", "Bottom Signal", "Top"`

SetNetVisible (Layout Editor)

Shows and highlights or hides specified nets in the Layout Editor. Multiple nets can be highlighted or hidden at once.

UI Access	On the **Nets** tab of the **Nets** window, highlight and right click on one or more nets. Click **Hide** or any of the **Show** options.
Parameters	

Name	Type	Description
<*requiredParameterKeyword*>	String	Required first parameter which must have the value "Name:Args".
<*requiredNameKeyword*>	String	Required second parameter which must have the value "Name:=". Must be the first parameter for each net.
<*netName*>	String	Name of the net
<*requiredFlagKeyword*>	String	Required fourth parameter which must have the value "Vis:=". Must be the third parameter for each net.
<*visibilityFlag*>	Boolean	True if the net should be visible. False if it should be hidden.

Return Value	None.
Python Syntax	SetNetVisible (["Name:Args", "Name:=", "<*netName*>", "Vis:=", <*highlightFlag*>, ...])

Python Example	`oEditor.SetNetVisible(`
	`  [`
	`  "NAME:Args",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Name:=", "Net_1",
    "Vis:=", True
  ])
oEditor.SetNetVisible(
  [
    "NAME:Args",
    "Name:=", "<NO-NET>",
    "Vis:=", False,
    "Name:=", "Ground",
    "Vis:=", False,
    "Name:=", "Chassis",
    "Vis:=", True
  ])
```

StitchLines (Layout Editor)

Use: Stitch together connected or crossing lines into polygons and lines.

Command: StitchLines

Syntax: StitchLines Array("NAME:elements", <"line">, …)

Return Value: None

Parameters: <"line">

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: text

Description: line name.

Example:

```
oDesign = oProject.SetActiveDesign("HFSS3D1")
oEditor = oDesign.SetActiveEditor("Layout")
oEditor.StitchLines ["NAME:elements", "line_1", "line_2", "line_3"]
```

Subtract (Layout Editor)

Use: Causes boolean subtracting of one or more *primitive* (polygons, rectangles, lines, or circles) object(s) from another one.

Syntax: **Subtract** Array ("NAME: *primitives*",

<object_name>, // Primitive to subtract from

<object_name>, // 1nd primitive to subtract, if any

…) // etc

Example:

```
oEditor.Intersect Subtract ("NAME:primitives", "circle_0", "rect_2")
```

ToggleViaPin (Footprint Editor)

Use: Selected pins are changed to vias. Selected vias are changed to pins.

Syntax: ToggleViaPin <NAME:elements", <object_name> ... // objects to be toggled

Return Value: None.

Parameters: <object_name>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: <String>

Example: `oEditor.ToggleViaPin ["NAME:elements", "via_195"]`

Ungroup (Layout Editor)

Use: Reverses a group operation; deletes a coordinate system but leaves the primitives.

Command: None.

Syntax: oEditor.Ungroup Array("NAME:elements", <CS1>, <CS2>, …)

Return Value: None.

Example: `oEditor.Ungroup ["NAME:elements", "CS_7"]`

Unite (Layout Editor)

Use: Causes Boolean uniting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: **Unite** Array ("NAME: *primitives*",

```
<object_name>, // 1st primitive name
<object_name>, // 2nd primitive, if any
…) // etc
```

Example:

```
oEditor.Unite ["NAME:primitives", "circle_0", "rect_2"]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UnselectAll (Layout Editor)

Use: Unselect all active selections in the layout editor.

Command: Edit > Unselect All

Syntax: UnselectAll()

Parameters: None

Example: `oLayout.UnselectAll()`

ZoomToFit (Layout Editor)

Zoom the current layout to fit the contents of the currently visible page

UI Access	**View** > **Fit All** or the **Fit All** button on the View ribbon.
Parameters	None.
Return Value	None.

Python Syntax	ZoomToFit()

Python Example	`oDesign = oProject.SetActiveDesign("projectName")`
	`oEditor = oDesign.SetActiveEditor("Layout")`
	`oEditor.ZoomToFit()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Symbol Editor Scripts

Symbol editor script commands are accessed with a definition editor.

```
oDefinitionEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "MySymbol")
```

The symbol editor script commands are listed below.

ChangeProperty (Symbol Editor)

CreateCircle (Symbol Editor)

CreateLine (Symbol Editor)

CreatePin (Symbol Editor)

CreatePolygon (Symbol Editor)

CreateRectangle (Symbol Editor)

CreateText (Symbol Editor)

Delete (Symbol Editor)

FlipHorizontal (Symbol Editor)

FlipVertical (Symbol Editor)

GetProperties (Symbol Editor)

GetPropertyValue (Symbol Editor)

Move (Symbol Editor)

Paste (Symbol Editor)

RemovePort (Symbol Editor)

Rotate (Symbol Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SelectAll (Symbol Editor)

SetPropertyValue (Symbol Editor)

Undo (Symbol Editor)

ZoomToFit (Symbol Editor)

AddLevel (Symbol Editor)

Adds a graphics level to the symbol.

UI Access	**Add Level** in the **View** > **Levels** dialog box
Parameters	Name | Type | Description <level> | Integer | The new level, an integer greater than 1.
Return Value	Integer, the previous active level. The current active level is the newly added level.

Python Syntax	oDefinitionEditor.AddLevel(<*level*>)

Python Example	`oProject = oDesktop.SetActiveProject("BPF_Microstrip_Resonant_Interdigitated")`
	`oDefinitionEditor = oProject.SetActiveDefinitionEditor("Sym-bolEditor","ExampleLibraries:load_power")`
	`oDefinitionEditor.AddLevel(2)`

AlignHorizontal (Symbol Editor)

Align items horizontally.

UI Access	**Draw** > **Align Horizontal**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><Selections></td><td>Array</td><td>["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify]]</td></tr><tr><td><AlignParameters></td><td>Array</td><td>["NAME:AlignParameters", "Disconnect:=", bool // [Opt=0] Should wires disconnect "Rubberband:=", bool] // [Opt=1] Should wires stair-case] Note: Alignment occurs relative to the first item in ids</td></tr></table>
Return Value	Value

Python Syntax	AlignHorizontal (<Selections>,<AlignParameters>)
Python Example	oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "Simplorer Elements\basic Elements\Circuit\PassiveElements:R") oSymbolEditor.AlignHorizontal (["NAME:Selections",_ "Selections:=", ["SymObj@11"]], 0)

AlignVertical (Symbol Editor)

Aligns items vertically.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Draw** > **Align Vertical**		
Parameters	Name	Type	Description
	<*Selections*>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify]]
	<*Align Parameters*>	Array	["NAME:AlignParameters", "Disconnect:=", bool // [Opt=0] Should wires disconnect "Rubberband:=", bool] // [Opt=1] Should wires staircase] Note: Alignment occurs relative to the first item in ids.
Return Value	None		

Python Syntax	AlignVertical (<*Selections*>, <*AlignParameters*>)
Python Example	oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "Simplorer Elements\basic Elements\Circuit\PassiveElements:R") oSymbolEditor.AlignVertical(["NAME:Selections","Selections:=", ["SymObj@11"]], 0)

BringToFront (Symbol Editor)

Changes the drawing for the symbol so that the specified objects are drawn on top of other overlapping objects.

UI Access	**Draw** > **Bring to Front**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Name*></td><td>String</td><td>"NAME:Selections"</td></tr><tr><td><*Selections*></td><td>Array</td><td>List of objects (strings) to bring to the front</td></tr></table>
Return Value	None

Python Syntax	BringToFront([<*Name*>, <*Selections*>])

Python Example	`oDefinitionEditor.BringToFront (`
	`  ["NAME:Selections",`
	`  "Selections:=", [ "SchObj@10"]`
	`  ])`

ChangeProperty (Symbol Editor)

Use: Changes to properties are scripted using the **ChangeProperty** command. This command can be executed by the **oEditor** to change editor properties, by the **oDesign** to change design level properties, and by the **oProject** to change project level properties. The command can be used to create, edit, and/or remove properties. In Circuit, only Variable and Separator properties can be deleted.

Command: None

Syntax: ChangeProperty Array("Name:AllTabs", <PropTabArray>, <PropTabArray>, ...)

Return Value: None

Parameters: <PropTabArray>

Array("Name:<PropTab>",

<PropServersArray>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<NewPropsArray>,
<ChangedPropsArray>,
<DeletedPropsArray>)

<PropServersArray>
Array("Name:PropServers", <PropServer>,
<PropServer>, ...)

<NewPropsArray>
Array("Name:NewProps", <PropDataArray>,
<PropDataArray>,...)

<ChangedPropsArray>
Array("Name:ChangedProps",<PropDataArray>,
<PropDataArray>, ...)

<DeletedPropsArray>
Array("Name:DeletedProps", <PropName>,
<PropName>, ...)
OR (for PropDisplay deletions only)
Array("Name:DeletedProps",<PropDataArray>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

<PropDataArray>, ...)

<PropDataArray>
Array("NAME:<PropName>",
"PropType:=", <PropType>,
"NewName:=", <string>,
"Description:=", <string>,
"Callback:=", <string>,
"NewRowPosition:=", <int>,
"ReadOnly:=", <bool>,
"Hidden:=", <bool>,
<PropTypeSpecificArgs>)OR (for PropDisplays only)
Array("Name:<PropName>",<PropDisplayData>)

<PropDisplayData>
for adding, changing, deleting PropDisplays
<PropDisplayAttributes>
for changing PropDisplays only
<PropDisplayNewAttributes>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<PropDisplayAttributes>

Layer & Location only used for PropDisplays in layout

For adding PropDisplays, this will add a single PropDisplay with attributes as shown; if an attribute is missing, a default value will be assigned. Adding PropDisplay to schematic with attributes that are identical to one already existing there will fail without an error message.

For deleting PropDisplays, these attributes are used to identify an existing PropDisplay to delete. If there doesn't exist a PropDisplay that matches the given attributes, then nothing will be deleted. If multiple PropDisplays match the given attributes, then all of them will be deleted. If an attribute is missing, then all PropDisplays match that missing attribute. For example, if Layer is missing, then PropDisplays on all layers that match the remaining given attributes will be deleted.

For changing PropDisplays, these attributes are used to identify an existing PropDisplay to change. If no PropDisplay matching the attributes is found, no changes will be made. If multiple PropDisplays match the attributes, all of them will be changed. If an attribute is missing, it matches all PropDisplays. For example, to change the format of PropDisplays that are on the bottom, but have any layer, style or format to show the name only, this command should have Location set to "Bottom" and all other attributes omitted.

"Format:=", <PropDisplayType>,

"Location:=", <PropDisplayLocation>,

"Layer:=", <string>,

"Style:=", <string>

<PropDisplayNewAttributes>

NewLayer & NewLocation only used for PropDisplays in layout

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

For changing PropDisplays, these attributes are used to identify which attributes to change and what the new value is. If the attribute should not be changed, the corresponding entry should be omitted.

"NewName:=", <string>,

"NewFormat:=", <PropDisplayType>,

"NewLocation:=", <PropDisplayLocation>,

"NewLayer:=", <string>,

"NewStyle:=", <string>

<PropDisplayType>

Type: string

Identifies the format of PropDisplay.

"Name"

"Value"

"NameAndValue"

"EvaluatedValue"

"NameAndEvaluatedValue"

<PropDisplayLocation>

Type: string

Identifies where PropDisplay is located with respect to object

"Left"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Top"

"Right"

"Bottom"

"Custom"

<PropType>

Type: string

Identifies the type of property when a new property is added. In Circuit, only separator properties and variable properties can be added.

"SeparatorProp"

"VariableProp"

"TextProp"

"NumberProp"

"ValueProp"

"CheckboxProp"

"MenuProp"

"PointProp"

"VPointProp"

"ButtonProp"

NewName

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Specify the new name of a property if the property's name is being edited. In Circuit, the name can only be changed for separators and variables.

Description

Specify a description of the property. In Circuit, the description can only be changed for separators and variables.

Callback

Specify the name of the script callback to be run when the property value is changed.

NewRowPosition

Used to reorder rows in the **Property** dialog box. In Circuit, this only applies to the **Project>Project Variables** panel and the **Designer>DesignProperties** panel. Specify the new zero-based row index of the variable or separator.

ReadOnly

Used to mark a property as "read only" so it can not be modified. In Circuit, this flag can only be set for variables and separators.

Hidden

Used to hide a property so it can not be viewed outside of the **Property** dialog box. In Circuit, this flag can only be set for variables and separators.

<PropTypeSpecificArgs>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SeparatorProp: no arguments

TextProp: "Value:=", <string>

NumberProp: "Value:=", <double>

ValueProp: "Value:=", <value>

CheckboxProp: "Value:=", <bool>

MenuProp: "Value:=", <string>

PointProp "X:=", <double>, "Y:=", <double>

VPointProp: "X:=", <value>, "Y:=", <value>

Material Button: "Material:=", <string>

Color Button: "R:=",<int>,"G:=",<int>,"B:=",<int>

Transparency Button: "Value:=", <double>

<PropTypeSpecificArgs> for MenuProps

Syntax for NewProps array: "AllChoices:=",

<"choice1,choice2,..."> or <Array("choice1" "choice2", ...)>,

"Value:=", <string>

Syntax for ChangedProps array: "Value:=", <string>

<PropTypeSpecificArgs> for VariableProps

Syntax:

"Value:=", <value>, <OptimizationFlagsArray>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<TuningFlagsArray>, <SensitivityFlagsArray>,

<StatisticsFlagsArray>

Parameters:

<OptimizationFlagsArray>

Array("NAME:Optimization",

"Included:=", <bool>,

"Min:=", <value>,

"Max:=", <value>)

<TuningFlagsArray>

Array("NAME:Tuning",

"Included:=", <bool>,

"Step:=", <value>,

"Min:=", <value>,

"Max:=", <value>)

<SensitivityFlagsArray>

Array("NAME:Sensitivity",

"Included:=", <bool>,

"Min:=", <value>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Max:=", <value>,

"IDisp:=", <value>)

<StatisticsFlagsArray>

Array("NAME:Statistical",

"Included:=", <bool>,

"Dist:=", <Distribution>,

"StdD:=", <value>,

"Min:=", <value>,

"Max:=", <value>,

"Tol:=", <string>)

<Distribution>

Type: string

Value should be **"Gaussian"** or **"Uniform"**

StdD

Standard deviation.

Min

Low cut-off for the distribution.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Max

High cut-off for the distribution.

Tol

Tolerance for uniform distributions. Format is **"<int>%"**.

Example: **"20%"**.

CloseEditor (Symbol Editor)

Closes the specified editor.

Command: None

Syntax: CloseEditor (VARIANT ptDelta)

Return Value: None

Python Syntax	CloseEditor()
Python Example	`oDefinitionEditor.CloseEditor()`

Copy (Symbol Editor)

Copy items for pasting.

*Command:***Edit** > **Copy**

Syntax: Copy Array("NAME:Selections", "Selections:=", Array(<selections>))

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None.

Parameters: <selections>

Type: Comma-separated list of strings

Identifiers for each selected item to be copied, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

CreateArc (Symbol Editor)

Draws an arc in the symbol editor

Command: None

Syntax: CreateArc

Return Value: None

Python Syntax	CreateArc()
Python Example	oDefinitionEditor.CreateArc (["NAME:ArcData", "X:=", -0.004318, "Y:=", -0.00127, "Radius:=", 0.00297299377732279, "StartAng:=", 1.9195673303788, "EndAng:=", 3.32144615338227, "Id:=", 10], ["NAME:Attributes", "Page:=", 1])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateCircle (Symbol Editor)

Draws a circle in the symbol editor

Command: None

Syntax: CreateCircle

Return Value: None

Python Syntax	CreateCircle()

Python Example	`oDefinitionEditor.CreateCircle`
	`(["NAME:CircleData",`
	`"X:=", -0.004572, "Y:=", -0.000508,`
	`"Radius:=", 0.001778, "Id:=", 12],`
	`["NAME:Attributes", "Page:=", 1])`

CreateImage (Symbol Editor)

Create picture from a BMP or GIF file.

*Command:***Draw>Image**

Syntax: CreateImage(

Return Value: None.

Parameters: `Array("NAME:ImageData",`

"X1:=", double, // the image rectangle left X value, in meters

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Y1:=", double, // the image rectangle upper Y value, in meters

"X2:=", double, // the image rectangle right X value, in meters

"Y2:=", double, // the image rectangle lower Y value, in meters

"File:=", string, // the name of the BMP or GIF file

"DisplayBorder:=", bool, // true or false, whether to display the image border

"LineWidth:=", double, // the width of the image border, in meters

"Color:=", int, // the RBG value of the image border color

"Id:=", int), // Id for this item

Array("NAME:Attributes",

"Page:=", int) // [Opt=1] The page number (one-based)

CreateLine (Symbol Editor)

Draws a line in the symbol editor

Command: None

Syntax: CreateLine

Python Syntax	CreateLine()

Python Example	`oDefinitionEditor.CreateLine`
	`(["NAME:LineData", _Return Value: None`
	`"Points:=", [ "(0.001778, -0.001270)",_`
	`"(0.004572, 0.002794)", "(0.003048, 0.003556)"],_`
	`"Id:=", 14], ["NAME:Attributes", "Page:=",1])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreatePin (Symbol Editor)

Draws a pin in the symbol editor

Command: None

Syntax: CreatePin

Return Value: None

Python Syntax	CreatePin()
Python Example	`oDefinitionEditor.CreatePin (["NAME:PinData", "Name:=", "newpin3"], ["NAME:PinParams", "X:=", -0.00762, "Y:=", 0.00254, "Angle:=", 0, "Flip:=", false])`

CreatePolygon (Symbol Editor)

Draws a polygon in the symbol editor

Command: None

Syntax: CreatePolygon

Return Value: None

Python Syntax	CreatePolygon()
Python Example	`oDefinitionEditor.CreatePolygon (["NAME:PolygonData", "X1:=", -0.001524, "Y1:=", 0.000508,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"X2:=", 0.002286, "Y2:=", -0.002032, "Id:=", 18], ["NAME:Attributes", "Page:=", 1])

CreateRectangle (Symbol Editor)

Draws a rectangle in the symbol editor

Command: None

Syntax: CreateRectangle

Return Value: None

Python Syntax	CreateRectangle()
Python Example	oDefinitionEditor.CreateRectangle (["NAME:RectData", "X1:=", -0.001524, "Y1:=", 0.000508, "X2:=", 0.002286, "Y2:=", -0.002032, "Id:=", 18], ["NAME:Attributes", "Page:=", 1])

CreateText (Symbol Editor)

Draws text in the symbol editor

Command: None

Syntax: CreateText

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateText()
Python Example	oDefinitionEditor.CreateText(["NAME:TextData","X:=", 0.001524, "Y:=", 0.00127, "Text:=","My text", "Id:=", 20], ["NAME:Attributes", "Page:=", 1])

Cut (Symbol Editor)

Cut page selection.

Command: None

Syntax: Cut(PageNumber, Selections)

Return Value: None

Python Syntax	Cut(["NAME:Selections", "Selections:=",[<selections>]])
Python Example	oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "Simplorer Elements\basic Elements\Circuit\PassiveElements:R") oSymbolEditor.Cut (["NAME:Selections", "Selections:=", ["SchObj@11", "SchObj@3"]])

Delete (Symbol Editor)

Delete items - remove from the symbol.

*Command:***Edit>Delete**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: Delete Array("NAME:Selections", "Selections:=", Array(<selections>))

Return Value: None.

Parameters: <selections>

Type: Comma-separated list of strings

Identifiers for each selected item to be deleted, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

DisplayPinNames (Symbol Editor)

To turn on or off the display of pin names.

*Command:*None.

Syntax: DisplayPinNames <onoff>

Return Value: None.

Parameters: <onoff>

Type: Boolean

TRUE or FALSE to turn on or off the display of pin names.

Example:

```
Dim oSymbolEditor

oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")

oSymbolEditor.DisplayPinNames TRUE
```

ExportFile (Symbol Editor)

Export file of symbol.

UI Access	**Symbol** > **Export File**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*type*></td><td>String</td><td>Type of file to export; only "EMF" is supported currently.</td></tr><tr><td><*fileName*></td><td>String</td><td>Path, name and extension for the file to be saved.</td></tr></table>
Return Value	None.

Python Syntax	ExportFile(<*type*>, <*filename*>)

Python Example	`oDefinitionManager = oProject.GetDefinitionManager()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`oSymbolManager.ExportFile("EMF","Filepath\Filename.EMF")`

FlipHorizontal (Symbol Editor)

To flip the specified objects about the horizontal (x) axis.

*Command:***Draw>Flip Horizontal**

Syntax: FlipHorizontal Array("NAME:Selections", "Selections:=", Array(<selections>)), 0

Return Value: None.

Parameters: `<selections>`

Type: Comma-separated list of strings

Identifiers for each selected item to be flipped, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

FlipVertical (Symbol Editor)

To flip the specified objects about the horizontal (x) axis.

*Command:***Draw>Flip Vertical**

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: FlipVertical Array("NAME:Selections", "Selections:=", Array(<selections>)), 0

Return Value: None.

Parameters: <selections>

Type: Comma-separated list of strings

Identifiers for each selected item to be flipped, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

GetProperties (Symbol Editor)

Gets a list of all the properties belonging to a specific **PropServer** and **PropTab**. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetProperties(<PropTab>, <PropServer>)

Return Value: Variant array of strings – the names of the properties belonging to the prop server.

GetPropertyValue (Symbol Editor)

Gets the value of a single property. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: GetPropertyValue(<PropTab>, <PropServer>, <PropName>)

Return Value: String representing the property value.

GetVisibleLevels (Symbol Editor)

To determine the levels in the symbol.

*Command:*None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: GetVisibleLevels

Return Value: An array of integers, which are the levels defined for the particular symbol.

Parameters: None.

GridSetup (Symbol Editor)

Changes the settings on the symbol grid.

*Command:***Symbol>Grid Setup**

Syntax: GridSetup(

Return Value: None.

```
Parameters: Array("NAME:Options", _
        "MajorGrid:=", string, _ // Major grid size with units
        "Divisions:=", int, _ // Number of minor grid divisions
        "MajorColor:=", int, _ // RGB Color of major grid lines
        "MinorColor:=", int, _ // RGB Color of minor grid lines
        "ShowGrid:=", bool, _ // Should the grid be shown?
        "SnapToGrid:=", bool, _ // Should objects snap to grid?
        "BackgroundColor:=", int, _ // RGB Color of the background
        "SaveAsDefault:=", bool)) // Should these settings be the default for new schematics
Example: oEditor.GridSetup Array(
  "NAME:Options",
  "MajorGrid:=", "10mm",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "Divisions:=", 10,
  "MajorColor:=", 0x00ff0000, // Red
  "MinorColor:=", 0x0000ff00, // Green
  "ShowGrid:=", true,
  "SnapToGrid:=", true,
  "BackgroundColor:=", 0x00ffffff, // White
  "SaveAsDefault:=", false)
```

Move (Symbol Editor)

To move the specified objects in X and Y.

*Command:*None.

Syntax: Move Array("NAME:Selections", "Selections:=", Array(<selections>)), Array("NAME:FlipParameters", "xdelta:=", <xval>, "ydelta:=",<yval>)

Return Value: None.

Parameters: <selections>

Type: Comma-separated list of strings

Identifiers for each selected item to be moved, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

<xval>

Type: Real number

The X delta to move.

<yval>

Type: Real number

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The Y delta to move.

Pan (Symbol Editor)

Pans the display.

UI Access	**View** > **Pan**.
Parameters	Name: <*argSelections*> — Type: Array of doubles — Description: First double pans along x-axis; second double pans along y-axis.
Return Value	None.

Python Syntax	Pan (<*argSelections*>)
Python Example	`oDefinitionEditor.Pan([-0.0092653869663931, 0.00686839904222147])`

Paste (Symbol Editor)

To paste copied objects to a specified X and Y location.

*Command:***Edit>Paste**

Syntax: Paste Array("NAME:Attributes", "X:=", <xval>, "Y:=",<yval>)

Return Value: None.

Parameters: <xval>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: Real number

The X placement value.

<yval>

Type: Real number

The Y placement value.

Redo (Symbol Editor)

Redo the last operation

Command: Click **Edit>Redo**

Syntax: Redo

Return Value: None

Parameters: None

RemoveLevels (Symbol Editor)

Removes one or more graphics levels from a symbol.

UI Access	**Tools** > **Edit Libraries** > **Symbols**. From Symbol editor, click **View** > **Levels** > **Remove Level**.
Parameters	Name | Type | Description <*levels*> | Array of integers | The levels to be deleted.
Return Value	None.

Python Syntax	RemoveLevels (<*levels*>)
Python Example	`oSymbolEditor.RemoveLevels([3,4])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemovePort (Symbol Editor)

Selected pins are changed to vias. Selected edge terminal ports are removed.

UI Access	**Not Available**
Parameters	Name / Type / Description <*Name*> / String / "NAME:elements" <*ObjectName*> / String / Name of object to be removed. Multiple objects can be listed, each as a separate string
Return Value	None

Python Syntax	RemovePort ([<NAME:elements", <*ObjectName*> ...])
Python Example	`oEditor.RemovePort(["NAME:elements", "via_195", "Port1"])`

Rotate (Symbol Editor)

To rotate the specified objects about their common center.

*Command:***Draw>Rotate**

Syntax: Rotate Array("NAME:Selections", "Selections:=", Array(<selections>)), Array("NAME:FlipParameters", <angle>)

Return Value: None.

Parameters: <selections>

Type: Comma-separated list of strings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Identifiers for each selected item to be rotated, in the form SchObj@<id>, where id is the value of the SchematicID property for each element.

<angle>

Type: A comma-separated string and real number

Either "Angle:=", <rad>, where <rad> is the angle in radians, or "Degrees:=", <deg>, where <deg> is the angle in degrees.

Save (Symbol Editor)

Saves any changes in the current symbol to the project.

*Command:***Symbol>Update Project**

Syntax: Save

Return Value: None.

Parameters: None.

Example:

```
oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor", "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")
```

SelectAll (Symbol Editor)

Select all elements in the symbol editor.

Command: **None**.

Syntax: SelectAll()

Parameters: None

Example: `removedDefs = oDefinitionEditor.SelectAll()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SelectAll()
Python Example	`removedDefs = oSymbolManager.SelectAll()`

SendToBack (Symbol Editor)

Changes the drawing for the symbol so that the specified objects are drawn behind other overlapping objects.

Command: Draw > Send To Back

Syntax: SendToBack Array("NAME:Selections", "Selections:=", Array (<Object>, <Object>, ...))

Return Value: None

Parameters: **<Object>**
<string> // object to send to the back

Example: `oDefinitionEditor.SendToBack ["NAME:Selections", "Selections:=", ["SchObj@10"]]`

Python Syntax	SendToBack (["NAME:Selections", "Selections:=",[<Object>, <Object>, ...]])
Python Example	`oDefinitionEditor.SendToBack(["NAME:Selections",` `"Selections:=",["SchObj@10"]])`

SetActivelLevel (Symbol Editor)

Sets the active graphics level in the symbol editor.

UI Access	Activate the **Current** radio button for a level in the **View** > **Levels** dialog box
Parameters	Name | Type | Description <*level*> | Integer | Specifies the new active level

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	SetActiveLevel (<*level*>)

Python Example	`oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",`
	`  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")`
	`oldLevel = oSymbolEditor.SetActiveLevel(4)`

SetInitialLevels (Symbol Editor)

Sets one or more graphics levels as the default visible levels for a symbol. The levels submitted do not need to contain 1 or 0, which are always visible.

UI Access	**Tools** > **Edit Libraries** > **Symbols**. From Symbol editor, click **View** > **Levels**. Select the **Default** check box for desired levels.
Parameters	Name | Type | Description <*levels*> | Array of integers | The levels to be selected.
Return Value	None.

Python Syntax	SetInitialLevels (<*levels*>)
Python Example	`oSymbolEditor.SetInitialLevels([3,4])`

SetPropertyValue (Symbol Editor)

Set a property.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: **None**.

Syntax: SetPropertyValue(tab, item, propname, value)

Parameters: None

Example:

```
removedDefs = oDefinitionEditor.SetPropertyValue(string tab, // Tab with the property
string item, // Name of the object
string propname, // Name of the property
string value) // The new value of the prop
```

SetVisibleLevels (Symbol Editor)

Makes one or more graphics levels visible in the symbol editor.

UI Access	**Tools** > **Edit Libraries** > **Symbols**. From Symbol editor, click **View** > **Levels**. Select the **Visible** check box for the desired levels.
Parameters	Name | Type | Description <*levels*> | Array of integers | The levels to be made visible.
Return Value	None.

Python Syntax	SetVisibleLevels (<*levels*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oSymbolEditor.SetVisibleLevels([3,4])`

ToggleLevel (Symbol Editor)

Toggles display of the graphics level in the symbol editor.

UI Access	Check the **Visible** option for a level in the **View** > **Levels** dialog box.
Parameters	Name: *<level>* — Type: Integer — Description: 1 to turn on the graphics display, 2 to turn off the graphics display
Return Value	None

Python Syntax	ToggleLevel (<level>)

Python Example	`oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",`
	`  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")`
	`oSymbolEditor.ToggleLevel(2)`

ToggleViaPin (Symbol Editor)

Selected pins are changed to vias. Selected vias are changed to pins.

Syntax: ToggleViaPin <NAME:elements", <object_name> ... // objects to be toggled

Return Value: None.

Parameters: <object_name>

Type: <String>

Example: `oEditor.ToggleViaPin ["NAME:elements", "via_195"]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Undo (Symbol Editor)

Undo the last operation

Command: Edit>Undo

Syntax: Undo

Return Value: None

Parameters: None

Example: oDesign.Undo

Python Syntax	Undo()
Python Example	oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",_ "Simplorer Elements\basic Elements\Circuit\PassiveElements:R") oSymbolEditor.Undo()

ZoomArea (Symbol Editor)

To change the current graphics view area.

*Command:***View>Zoom Area**

Syntax: ZoomArea Array(<xpt>, <ypt>), Array(<xdelta>, <ydelta>)

Return Value: None.

Parameters: <xpt>

Type: Real number

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The X value of the zoom area box, in meters.

<ypt>

Type: Real number

The X value of the zoom area box, in meters.

<xdelta>

Type: Real number

The width of the zoom area box, in meters.

<ydelta>

Type: Real number

The height of the zoom area box, in meters.

Example:

```
oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",
  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")
oSymbolEditor.ZoomArea [0, 0], [0.004, 0.004]
```

ZoomIn (Symbol Editor)

To zoom in a standard amount to the current graphics view area.

*Command:***View>Zoom In**

Syntax: ZoomIn

Return Value: None.

Parameters: None.

Example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",
  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")
oSymbolEditor.ZoomIn
```

ZoomOut (Symbol Editor)

To zoom out a standard amount to the current graphics view area.

*Command:***View>Zoom Out**

Syntax: ZoomOut

Return Value: None.

Parameters: None.

Example:

```
oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",
  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")
oSymbolEditor.ZoomOut
```

ZoomPrevious (Symbol Editor)

To zoom the current graphics view area to the previous extents.

*Command:***View>Zoom Previous**

Syntax: ZoomPrevious

Return Value: None.

Parameters: None.

Example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",
  "Simplorer Elements\basic Elements\Circuit\Passive Elements:R")
oSymbolEditor.ZoomPrevious
```

ZoomToFit (Symbol Editor)

Zooms the current symbol to fit the contents of the currently visible page.

UI Access	**View** > **Fit All** or the **Fit All** button on the View ribbon.
Parameters	None.
Return Value	None.

Python Syntax	ZoomToFit()

Python Example	`oSymbolEditor = oProject.SetActiveDefinitionEditor("SymbolEditor",_`
	`"Simplorer Elements\basic Elements\Circuit\PassiveElements:R")`
	`oSymbolEditor.ZoomToFit()`

Footprint Editor Scripts

Footprint editor script commands are accessed with a definition editor.

```
oDefinitionEditor = oProject.SetActiveDefinitionEditor("FootprintEditor", "MyFootprint")
```

The footprint editor script commands are listed below.

AddLayer (Footprint Editor)

AddStackupLayer (Footprint Editor)

ChangeLayers (Footprint Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeOptions (Footprint Editor)
CreateCircle (Footprint Editor)
CreateCircleVoid (Footprint Editor)
CreateEdgePort (Footprint Editor)
CreateLine (Footprint Editor)
CreateLineVoid (Footprint Editor)
CreateMeasure (Footprint Editor)
CreatePolygonVoid (Footprint Editor)
CreatePin (Footprint Editor)
CreatePolygon (Footprint Editor)
CreateRectangle (Footprint Editor)
CreateText (Footprint Editor)
CreateVia (Footprint Editor)
Duplicate (Footprint Editor)
Edit (Footprint Editor)
EraseMeasurements (Footprint Editor)
FlipHorizontal (Footprint Editor)
FlipVertical (Footprint Editor)
GetAllLayerNames (Footprint Editor)
GetLayerInfo (Footprint Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProperties (Footprint Editor)

GetStackupLayerNames (Footprint Editor)

Intersect (Footprint Editor)

Move (Footprint Editor)

PageSetup (Footprint Editor)

RemoveLayer (Footprint Editor)

RemovePort (Footprint Editor)

Rotate (Footprint Editor)

SetPropertyValue (Footprint Editor)

Subtract (Footprint Editor)

ToggleViaPin (Footprint Editor)

Unite (Footprint Editor)

ZoomToFit (Footprint Editor)

AddLayer (Footprint Editor)

Add a layer in a footprint definition. This command can take all parameters available in ChangeLayers (Footprint Editor) but cannot be recorded.

UI Access	None		
Parameters	Name	Type	Description
	<LayerArray>	Array	An array that contains these parameters in order: • <Layername> • <Type> • <TopBottom>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		• <Color> • <Transparency> • <Pattern> • <VisFlag> • <Locked> • <Draw>
<LayerName>	String	Name of layer.
<Type>	String	Type of layer, such as "user", "signal", "dielectric", "soldermask", "solder-paste", "silkscreen", "assembly", "wirebond", "dielectric", "glue", "user", "Post-processing", or "outline".
<TopBottom>	String	Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
<Color>	Integer	Optional. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
<Transparency>	Integer	Optional. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
	<Locked>	Boolean	Optional. Whether the layer is locked.
	<Draw>	Integer	Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
Return Value	None		

Python Syntax	oEditor.AddLayer(["NAME:layers", <LayerName>", <Type>", <TopBottom>", <Color>, <Transparency>, <Pattern>, <VisFlag>, <Locked>, <Draw>,])

Python Example	`oEditor = oDesign.SetActiveEditor("Footprint")`
	`oEditor.AddLayer(`
	`  [`
	`  "NAME:layers",`
	`  "Name:=","signal",`
	`  "Type:=", "signal",`
	`  "Top Bottom:=", "neither",`
	`  "Color:=", 16711680,`
	`  "Transparency:=", 0,`
	`  "Pattern:=", 1,`
	`  "VisFlag:=", 127,`
	`  "Locked:=", False,`
	`  "DrawOverride:=", 0,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

AddStackupLayer (Footprint Editor)

Adds a stackup layer in a footprint definition. This command can take all parameters available in ChangeLayers (Footprint Editor) but cannot be recorded.

UI Access	None
Parameters	See below

Name	Type	Description

<LayerArray>	Array	An array that contains these parameters in order:
		• <Layername>
		• <Type>
		• <TopBottom>
		• <Color>
		• <Transparency>
		• <Pattern>
		• <VisFlag>
		• <Locked>
		• <Draw>
		• and an array with the parameters:
		  • <Thickness>
		  • <LowerElevation>
		  • <Material>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<LayerName>	String	Name of layer.
<Type>	String	Type of layer, such as "user", "signal", "dielectric", "soldermask", "solderpaste", "silkscreen", "assembly", "wirebond", "dielectric", "glue", "user", "Post-processing", or "outline".
<TopBottom>	String	Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
<Color>	Integer	Optional. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
<Transparency>	Integer	Optional. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
<Locked>	Boolean	Optional. Whether the layer is locked.
<Draw>	Integer	Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
<Thickness>	String	Thickness of the layer, contains the measurement and units.
<LowerElevation>	String	The elevation of the lower edge of the layer, contains the measurement and units.
<Material>	String	The material the layer is made of.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	oEditor.AddStackupLayers(["NAME:stackup layer",<LayerName>, <Type>, <TopBottom>, <Color>, <Transparency>, <Pattern>, <VisFlag>, <Locked>, <Draw>, ["NAME:sublayer", <Thickness>, <Lower-Elevation>, <Material>,]])

Python Example	`oEditor = oDesign.SetActiveEditor("Footprint")`
	`oEditor.AddStackupLayer(`
	`  [`
	`  "NAME:stackup layer",`
	`  "Name:=", "Cover",`
	`  "Type:=", "signal",`
	`  "Top Bottom:=", "neither",`
	`  "Color:=", 16711680,`
	`  "Transparency:=", 0,`
	`  "Pattern:=", 1,`
	`  "VisFlag:=", 127,`
	`  "Locked:=", False,`
	`  "DrawOverride:=", 0,`
	`  [`
	`  "NAME:Sublayer",`
	`  "Thickness:=", "0mil",`
	`  "LowerElevation:=", "20mil",`
	`  "Material:=", "gold",`
	`  ]`
	`  ])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeLayer (Footprint Editor)

Changes one or more attributes or parameters of a single layer.

> **Note:** To execute the command, **all** of the following parameters must be entered in the command line even if the user only intends to change one attribute.

<table>
<tr><td>UI Access</td><td colspan="3">Create or change a layer in the Edit Layers window.</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><LayerArray></td><td>Array</td><td>An array that contains these parameters in order:
<mode>
<Layername>
<ID>
<Type>
<TopBottom>
<Color>
<Transparency>
<Pattern>
<VisFlag>
<Locked>
<Draw>
<Subarray></td></tr>
<tr><td><mode></td><td>String</td><td>Type of stackup. Can be "Laminate","Multizone", or "Overlap" .</td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of layer.</td></tr>
<tr><td><ID></td><td>Integer</td><td>A numerical identification for the layer.</td></tr>
<tr><td><Type></td><td>String</td><td>Type of layer, such as "user", "signal", "dielectric", "soldermask", "solder-</td></tr>
</table></td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		paste", "silkscreen", "assembly", "wirebond", "dielectric", "glue", "user", "Post-processing", or "outline".
<TopBottom>	String	*Optional*. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".
<Color>	Integer	*Optional*. A code to indicate the color of the layer. It is an RGB code in the hex format *bbggrr*. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.
<Transparency>	Integer	*Optional*. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).
<Pattern>	Integer	*Optional*. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
<Locked>	Boolean	*Optional*. Whether the layer is locked.
<DrawOverride>	Integer	*Optional*. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
<Zones>		
<Subarray>	Array	*Optional*. An array that contains these parameters in order: • <Thickness> • <LowerElevation>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• <Roughness> • <BotRoughness> • <SideRoughness> • <Material> • <FillMaterial> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.
	<Thickness>	String	Thickness of the layer, contains the measurement and units.
	<LowerElevation>	String	The elevation of the lower edge of the layer, contains the measurement and units.
	<Roughness>	String	The roughness of the top of the layer, contains the measurement and units.
	<BotRoughness>	String	The roughness of the bottom of the layer, contains the measurement and units.
	<SideRoughness>	String	The roughness of the side of the layer, contains the measurement and units.
	<Material>	String	The material the layer is made of.
	<FillMaterial>	String	The material to use to fill in around geometry on the layer.
Return Value	None		

Python Syntax	oEditor.ChangeLayer(<LayerArray>)

Python Example	`oEditor = oDesign.SetActiveEditor("Footprint")`
	`oEditor.ChangeLayer(`
	`  ["NAME:stackup layer",`
	`  "Name:="  , "MQ",`
	`  "ID:="  , 20,`
	`  "Type:="  , "signal",`
	`  "Top Bottom:="  , "neither",`
	` "Color:=" , 6136303,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Transparency:="   , 60,
    "Pattern:="        , 1,
    "VisFlag:="        , 96,
    "Locked:="         , False,
    "DrawOverride:="   , 0,
    "Zones:="          , [],
      ["NAME:Sublayer",
        "Thickness:="        , "0.745um",
        "LowerElevation:="   , "737.155um",
        "Roughness:="        , "0um",
        "BotRoughness:="     , "0um",
        "SideRoughness:="    , "0um",
        "Material:="         , "MQ",
        "FillMaterial:="     , "FR4_epoxy"
      ]
    ]
  )
```

ChangeLayers (Footprint Editor)

Changes one or more attributes or parameters of a layer.

> **Note:** To execute the command, **all** of the following parameters must be entered in the command line even if the user only intends to change one attribute.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Create or change a layer in the Footprint editor with theEdit Layers window.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><LayerArray></td><td>Array</td><td>An array that contains these parameters in order:
<Layername>
<ID>
<Type>
<TopBottom>
<Color>
<Transparency>
<Pattern>
<VisFlag>
<Locked>
<Draw>
<Subarray>
<Neg></td></tr>
<tr><td><mode></td><td>String</td><td>Type of stackup. Can be "Laminate","Multizone", or "Overlap" .</td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of layer.</td></tr>
<tr><td><ID></td><td>Integer</td><td>A numerical identification for the layer.</td></tr>
<tr><td><Type></td><td>String</td><td>Type of layer, such as "user", "signal", "dielectric", "soldermask.", "solder-paste", "silkscreen", "assembly", "wirebond", "dielectric", "glue", "user", "Post-processing", or "outline".</td></tr>
<tr><td><TopBottom></td><td>String</td><td>Optional. Indicates whether the layer is the top or bottom. Acceptable values are "top", "bottom", or "neither".</td></tr>
<tr><td><Color></td><td>Integer</td><td>Optional. A code to indicate the color of the layer. It is an RGB code in the hex format bbggrr. For example, red (#0000ff) is 255, green (#00ff00) is 65280, and blue (#ff0000) is 16711680.</td></tr>
<tr><td><Transparency></td><td>Integer</td><td>Optional. A percentage that indicates the transparency of the layer between 0 (opaque) and 100 (completely transparent and not visible).</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<Pattern>	Integer	Optional. A code for the pattern of the layer: 0 is hollow,1 is solid, and the other patterns 3-8 are various hatch patterns.
<VisFlag>	Integer	Sets visibility for all objects on the layer using the following additive code: • 1 makes primitives (rectangles, circles, polygons) visible. • 2 makes lines (also known as paths or traces) visible. • 4 makes pads (from padstacks, aka vias) visible. • 8 makes holes (or vias) visible. • 16 makes components visible. • 32 makes the mesh overlay visible if they are plotted. • 64 makes the mesh for the background material visible. For example, a value of 25 (1+8+16) makes primitives, vias, and components visible. Use 127 to show everything and 0 to hide all.
<Locked>	Boolean	Optional. Whether the layer is locked.
<Draw>		Optional. Allows drawing on the layer as wireframe, outlines, or solid. Acceptable values are -1 for wireframe, 0 for normal, and 1 for solid.
<Subarray>	Array	An array that contains these parameters in order: • <Thickness> • <LowerElevation> • <Roughness> • <BotRoughness> • <SideRoughness> • <Material> • <FillMaterial> This array's placeholder does not appear in the syntax or examples below but instead has its contents listed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Thickness>	String	Thickness of the layer, contains the measurement and units.
	<LowerElevation>	String	The elevation of the lower edge of the layer, contains the measurement and units.
	<Roughness>	String	The roughness of the top of the layer, contains the measurement and units.
	<BotRoughness>	String	The roughness of the bottom of the layer, contains the measurement and units.
	<SideRoughness>	String	The roughness of the side of the layer, contains the measurement and units.
	<Material>	String	The material the layer is made of.
	<FillMaterial>	String	The material to use to fill in around geometry on the layer.
	<Neg>	Boolean	Whether the geometry on the layer is cut away from the layer.
Return Value	None		

Python Syntax

```
oEditor.ChangeLayers(
    [
        "NAME:layers",
        "Mode:=", "<mode>",
        [
            "NAME:pps"
        ],
        [
            "NAME:stackup layer",
            "Name:=", "<LayerName>",
            "ID:=", <ID>,
            "Type:=", "<Type>",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "Top Bottom:=", "<TopBottom>",
            "Color:=", <Color>,
            "Transparency:=", <Transparency>,
            "Pattern:=", <Pattern>,
            "VisFlag:=", <VisFlag>,
            "Locked:=", <Locked>,
            "DrawOverride:=", <Draw>,
                [
                "NAME:Sublayer",
                "Thickness:=", "<Thickness>",
                "LowerElevation:=", "<LowerElevation>",
                "Roughness:=", "<Roughness>",
                "BotRoughness:=", "<BotRoughness>",
                "SideRoughness:=", "<SideRoughness>",
                "Material:=", "<Material>",
                "FillMaterial:=", "<FillMaterial>"
            ],
            "Neg:=", <Neg>
        ]
    ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example

```
oEditor = oDesign.SetActiveEditor("Footprint")
oEditor.ChangeLayers(
  [
    "NAME:layers",
    "Mode:=", "Laminate",
    [
      "NAME:pps"
    ],
    [
      "NAME:stackup layer",
      "Name:=", "Cover",
      "ID:=", 10,
      "Type:=", "signal",
      "Top Bottom:=", "neither",
      "Color:=", 16711680,
      "Transparency:=", 0,
      "Pattern:=", 1,
      "VisFlag:=", 127,
      "Locked:=", False,
      "DrawOverride:=", 0,
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "NAME:Sublayer",
          "Thickness:=", "0mil",
          "LowerElevation:=", "20mil",
          "Roughness:=", "0um",
          "BotRoughness:=", "0um",
          "SideRoughness:=", "0um",
          "Material:=", "gold",
          "FillMaterial:=", "FR4_epoxy"
        ],
        "Neg:=", True
      ]
  ])
```

ChangeOptions (Footprint Editor)

Changes options for an existing layout. Does not change global options specified in the registry. Only those options being changed need to be specified. Options not specified are not affected.

UI Access	Access via scripting only. No direct UI command.
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"NAME:Options"	String	Required header for the options array. Must be the first item in the list.
	<OptionData>	Mixed	Option name and value pair. Value can be: **String** (e.g., `"SnapMode:=", "Fine"`) **Float** (e.g., `"DrawingResolution:=", 64.0`) **Array of floats** (e.g., `"PageExtent:=", [-0.1, -0.1, 0.1, 0.1]`)
Return Value	None		

Python Syntax	oEditor.ChangeOptions(["NAME:options", <OptionData>, …])

Python Example	`oEditor.ChangeOptions([`
	`  "NAME:Options",`
	`  "useFixedDrawingResolution:=", False,`
	`  "DrawingResolution:=", 64,`
	`  "ShowGrid:=", True,`
	`  "MajorSize:=", 1.0,`
	`  "MinorSize:=", 0.1,`
	`  "PageExtent:=", [-0.1, -0.1, 0.1, 0.1],`
	`  "DefaultToSketchMode:=", False,`
	`  "NetColor:=", 16777152`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

An error will be returned if this script command is not used at the top-level hierarchy.

- Global layout defaults are stored in the registry (Layout\Preferences\)
- Names of registry items were selected for backwards compatibility and are consistent with adsn, technology file, and scripting naming
- Local options are both script-able and undo-able
- Global options are neither script-able nor undo-able

Option	Description	Installed default	Registry, Scripting Names, Data Type
Arc Drawing Resolution	Controls drawing of segments for arcs - either dynamic and based on current zoom or fixed to specified value	Dynamic	"useFixedDrawingResolution"[1] "DrawingResolution" [2]
Major and Minor Grid lines	Spacing, color, and visibility	The grid is displayed. Spacing based on current default length units. Major grid lines are 10 minor grid lines apart. The line colors are light blue grays, with major grid lines being darker.	"MajorColor" [3] "MinorColor" [3] "MajorSize" [2] "MinorSize" [2] "ShowGrid" [1]
Drawing Extent	Specifies size and coordinates of the layout	Lower left of the layout is (-0.1, -0.1) and the upper right is (0.1, 0.1)	"PageExtent" [4]
Background	Color of the layout background	white	"background color" [3]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Color			
Sketch Mode	Fill patterns and center lines are not drawn.	off	"DefaultToSketchMode" [1]
Solid Mode	Objects are filled in with solid color.	off	"fillMode" [1]
Draw Connection Points	Display pin symbols in the layout.	off	"show connection points" [1]
Draw Rats	Display lines indicating missing physical connections in nets	on	"draw rats" [1]
Label Vertices	Display property text labeling the vertices of selected polygons and lines.	off	"display vertex labels" [1]
Net Color	Color used for highlighting nets	light yellow	"NetColor" [3]
Rectangle Description	Specifies if rectangles are described with two points or center point, width, and height	2 points	"rectangle description" [5]
Default Hole Size	Default size for actual holes in a PC board	25 mil	"defaultholesize" [2]
Align Microwave Components	Snaps components on entry.	on	"snaptoport" [1]
Symbol Footprint Scaling	Used to size symbol footprints placed in layout.	Based on data extent and current length units.	"sym footprint scaling" [6]
Primary and Secondary Selection Colors	Colors used to indicate the first item selected and other items currently selected.	Primary color is red, secondary color is a darker red	"primary selectioncolor" [3] "secondary selection color" [3]
Preview Selection	Highlight the object that would be selected with a left mouse button	off	"preview selection" [1]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	click in the current location.		
Snapping Options	Choose snapping sources and targets and the maximum distance (in pixels) for snapping to occur. Constrain edits to grid if desired. **Note:** Off-grid objects are ignored if ConstrainToGrid is asserted.	Snap distance is 20 pixels. Snapping sources are vertex, edge center, object center, & elec. conn. Snapping targets are vertex, edge center, object center, elec. conn, & grid. Edits are constrained to grid.	"PixelSnapTolerance"[7] "SnapTargetVertex_on"[1] "SnapTargetEdgeCenter_on"[1] "SnapTargetObjCenter_on"[1] "SnapTargetEdge_on"[1] "SnapTargetElecConnection_on"[1] "SnapTargetIntersection_on"[1] "SnapTargetGrid_on"[1] "SnapSourceVertex_on"[1] "SnapSourceEdgeCenter_on"[1] "SnapSourceObjCenter_on"[1] "SnapSourceEdge_on"[1] "SnapSourceElecConnection_on"[1] "ConstrainToGrid"[1]
Always Show Merge Layers Dialog	Controls display of the layer merging dialog.	off – Only show dialog when the user must be involved.	"AlwaysShowLayerMergeDlg"[1]
Rotation Increment	During rotation, objects are rotated by multiples of this amount.	Based on current angle units.	"anglesnap" [2]
Allow Drag on First Click	Selection and move with one left mouse button click.	false	"AllowDragOnFirstClick" [1]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

[1] Boolean

[2] Integer

[3] String

[4] [float, float, float, float]

[5] Enumeration

CloseEditor (Footprint Editor)

Closes the active definition editor.

UI Access	None
Parameters	None
Return Value	None

Python Syntax	CloseEditor()
Python Example	`oEditor.CloseEditor()`

CreateCircle (Footprint Editor)

Draws a circle in the footprint editor.

UI Access	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters

Name	Type	Description
<*NAME:Contents*>	String	Object descriptor header.
<*CircleGeometry*>	Array	Array of geometric properties for the circle.
<*LayerName*>	String	Name of the layer.
<*Lw*>	Integer	Border line width.
<*X*>	Integer	Center X coordinate.
<*Y*>	Integer	Center Y coordinate.
<*R*>	Integer	Radius.

Return Value

Object name (string)

Python Syntax

CreateCircle([<*NAME:Contents*>, <*CircleGeometry*>, [<*LayerName*>, <*Lw*>, <*X*>, <*Y*>, <*R*>]])

Python Example

```
oEditor.CreateCircle(
  [
       "NAME:Contents",
       "CircleGeometry:=", [
       "LayerName:=", "TopLayer",
       "Lw:=", "0.2mm",
       "X:=", "10mm",
       "Y:=", "15mm",
       "R:=", "5mm"
    ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
   ]
)
```

CreateCircleVoid (Footprint Editor)

Creates a circle void and adds it to a particular parent primitive.

UI Access	Available only through scripting.
Parameters	

Name	Type	Description
<NAME:Contents>	String	Header string identifying the array contents.
<Owner>	String	Name of the parent primitive object to which the void is added.
<Circle VoidGeometry>	Array	Contains geometric parameters for the void. Format is assumed to match `CreateCircle` geometry array.
<*LayerName*>	String	Name of the layer.
<*Lw*>	Integer	Border line width.
<*X*>	Integer	Center X coordinate.
<*Y*>	Integer	Center Y coordinate.
<*R*>	Integer	Radius

Return Value	String (object name)

Python Syntax	oEditor.CreateCircleVoid([<NAME:Contents>, <Owner>, <Circle VoidGeometry>])
Python Example	

```
oEditor = oDesign.SetActiveEditor("Footprint")
oEditor.CreateCircleVoid(
  ["NAME:Contents",
    "Owner:=", "Pad1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Circle VoidGeometry:=",
      ["LayerName:=", "Top",
      "lw:=", "0.1mm",
      "x:=", "0mm",
      "y:=", "0mm",
      "r:=", "0.5mm"]
  ]
)
```

CreateEdgePort (Footprint Editor)

Creates a new edge port using one or more specified edges.

UI Access	From **Draw**, select **Port** > **Create**.

Parameters

Name	Type	Description
<NAME:Contents>	String	Required header string identifying the parameter block.
<edge>	Array	Array describing a single edge. Each entry contains: `et`, `prim`, `edge`, `sel`, `layer`, `sx`, `sy`, `ex`, `ey`, `h`, `rad`
<et>	String	Edge type: "pe" for primitive, "pse" for via.
<prim>	String	Primitive name (used if `et` = "pe").
<edge>	Integer	Edge number on the primitive.
<sel>	String	Via name (used if `et` = "pse").

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*layer*></td><td>Integer</td><td>ID of the layer of the via pad.</td></tr><tr><td><*sx*>, <*sy*></td><td>Integer</td><td>X, Y location of the start point of the arc.</td></tr><tr><td><*ex*>, <*ey*></td><td>Integer</td><td>X, Y location of the end point of the arc.</td></tr><tr><td><*h*></td><td>Integer</td><td>Height of the arc (0 for straight edge).</td></tr><tr><td><*rad*></td><td>Integer</td><td>Arc size in radians (0 for straight edge).</td></tr><tr><td><*external*></td><td>Boolean</td><td>True if the port is external; false if internal.</td></tr></table>
Return Value	Returns the name of the created edge port. Returns an empty string if creation fails.
Python Syntax	CreateEdgePort([<NAME:Contents>, <edge>, <external>])

Python Example	`oEditor.CreateEdgePort([`
	`  "NAME:Contents",`
	`  "edge:=", [`
	`  "et:=", "pe",`
	`  "prim:=", "Prim1",`
	`  "edge:=", 2`
	`  ],`
	`  "external:=", True`
	`])`

CreateLine (Layout Editor)

Creates a line primitive object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>None</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><NAME:Contents></td><td>String</td><td>Object descriptor header.</td></tr>
<tr><td><LineGeometry></td><td>Array</td><td>Array of geometric properties for the line.</td></tr>
<tr><td><LayerName></td><td>String</td><td>Name of the layer.</td></tr>
<tr><td><Lw></td><td>Integer</td><td>Line width.</td></tr>
<tr><td><EndStyle></td><td>Integer</td><td>Line end style: 0 = flat end, 1 = extended end, 2 = round end.</td></tr>
<tr><td><JoinStyle></td><td>Integer</td><td>Line join style: 0 = unmitered join, 1 = mitered join, 2 = optimally mitered join, 3 = radially mitered join, 4 = chord mitered join.</td></tr>
<tr><td><VertexSequence></td><td>Array</td><td>Sequence of vertex coordinates: "n:=" count, followed by "x0:=", "y0:=", "x1:=", "y1:=", etc.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Object name (string)</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>CreateLine([<NAME:Contents>, <LineGeometry>, [<LayerName>, <Lw>, <EndStyle>, <JoinStyle>, <VertexSequence>]])</td></tr>
<tr><td>Python Example</td><td>

```
oEditor.CreateLine(
  [
    "NAME:Contents",
    "LineGeometry:=", [
      "LayerName:=", "TopLayer",
      "Lw:=", 2,
      "EndStyle:=", 1,
```

</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "JoinStyle:=", 2,
        "n:=", 2,
        "x0:=", 10,
        "y0:=", 15,
        "x1:=", 25,
        "y1:=", 30
      ]
    ]
)
```

CreateLineVoid (Footprint Editor)

Use: Creates a line void and adds it to a specified as parameter parent primitive. Returns the name of the newly created object.

Syntax: CreateLineVoid<line_void_description>

<line_void_description>:

Array("NAME:Contents",

"owner:=", <object_name>, // parent primitive name

"line voidGeometry:=", <line_geometry>) // definition

CreatePolygonVoid (Footprint Editor)

Use: Creates a polygon void and adds it to a specified as parameter parent primitive.

Syntax: CreatePolygonVoid<polygon_void_description>

<polygon_void_description>:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
Array("NAME:Contents",
"owner:=", <object_name>, // owner name
"poly voidGeometry:=", <polygon_geometry>) // definition
```

Return Value: Returns the name of the newly created object.

CreatePin (Footprint Editor)

Creates a pin in the footprint editor.

UI Access	None.
Parameters	None.
Return Value	None.

Python Syntax	CreatePin()
Python Example	`oEditor.CreatePin()`

CreatePolygon (Footprint Editor)

Creates a polygon primitive object.

UI Access	None
Parameters	

Name	Type	Description
<NAME:Contents>	String	Object descriptor header.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<LineGeometry>	Array	Array of geometric properties for the polygon, consisting of the following parameters in order: • *<LayerName>* • *<Lw>* • *<N>* • *<X0>* • *<Y0>* • *<X1>* • *<Y1>* • *<X2>* • *<Y2>* • *<X3>* • *<Y3>*
<LayerName>	String	Name of the layer.
<Lw>	Integer	Border line width.
<N>	Integer	Number of polygon vertices.
<X0>	Double	X-coordinate of vertex 0.
<Y0>	Double	Y-coordinate of vertex 0.
<X1>	Double	X-coordinate of vertex 1.
<Y1>	Double	Y-coordinate of vertex 1.
<X2>	Double	X-coordinate of vertex 2.
<Y2>	Double	Y-coordinate of vertex 2.
<X3>	Double	X-coordinate of vertex 3.
<Y3>	Double	Y-coordinate of vertex 3.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	Object name (string)

Python Syntax	CreatePolygon([<*NAME:Contents*>, [<*LineGeometry*>]])

Python Example	`oEditor.CreatePolygon(`
	`  [`
	`  "NAME:Contents",`
	`  "LineGeometry:=", [`
	`  "LayerName:=", "TopLayer",`
	`  "Lw:=", 2,`
	`  "n:=", 4,`
	`  "x0:=", 10,`
	`  "y0:=", 15,`
	`  "x1:=", 25,`
	`  "y1:=", 30,`
	`  "x2:=", 40,`
	`  "y2:=", 20,`
	`  "x3:=", 15,`
	`  "y3:=", 5`
	`  ]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

CreateRectangle (Footprint Editor)

Use: Draws a rectangle in the footprint editor

Command: None

Syntax: CreateRectangle

Return Value: None

CreateText (Footprint Editor)

Use: Draws text in the footprint editor

Command: None

Syntax: CreateText

Return Value: None

CreateVia (Footprint Editor)

Use: Creates a via in the footprint editor

Command: None

Syntax: CreateVia

Return Value: Name of the created via.

Duplicate (Footprint Editor)

Use: Duplicates footprint objects.

Command: The specified objects are duplicated by the given count with the specified offset.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: Duplicate Array("NAME:options", "count:=", <count data>), Array("NAME:elements", <element names>), Array(<x>, <y>)

Return Value: None

Parameters: <count data> is an integer

<element names> is one or more strings with the names of footprint objects

<x> is the x value for the offset

<y> is the y value for the offset

Edit (Footprint Editor)

Use: Edits an object in the footprint editor

Command: None

Syntax: Edit

Return Value: None

EraseMeasurements (Footprint Editor)

Use: Causes erasing of ALL measurements currently present.

Command: None.

Syntax: EraseMeasurements

Return Value: None.

Parameters: None.

FlipHorizontal (Footprint Editor)

Use: Flips the selected object horizontally

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: None

Syntax: FlipHorizontal

Return Value: None

FlipVertical (Footprint Editor)

Use: Flips the selected object vertically

Command: None

Syntax: FlipVertical

Return Value: None

GetAllLayerNames (Footprint Editor)

Use: Informational.

Command: None.

Syntax: GetAllLayerNames

Return Value: Array of strings which are the names of all layers in the layout, blackbox, or footprint.

Parameters: None.

GetLayerInfo (Footprint Editor)

Retrieves information on a specified layer in the Footprint editor.

<table>
<tr><td>UI Access</td><td colspan="3">None</td></tr>
<tr><td>Parameters</td><td colspan="3"><table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><LayerName></td><td>String</td><td>Name of the layer</td></tr></table></td></tr>
<tr><td>Return Value</td><td colspan="3">An array of strings, as follows:</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• Type: typename
	• TopBottomAssociation: "Top"|"Neither"|"Bottom"|"Template"|"Invalid"
	• Color: integer [representing rgb in hex]
	• IsVisible: "true"|"false"
	• IsVisible: "true"|"false". This is true if any these types of objects are visible:
	  • IsVisibleShape: "true"|"false"[for stackup layer only]
	  • IsVisiblePath: "true"|"false"[for stackup layer only]
	  • IsVisiblePad: "true"|"false"[for stackup layer only]
	  • IsVisibleHole: "true"|"false"[for stackup layer only]
	  • IsVisibleComponent: "true"|"false"[for stackup layer only]
	• IsLocked: "true"|"false"
	• LayerId: integer [the ID for the layer]
	• IsLocked: "true"|"false"
	• LayerId: integer [the ID for the layer]
	The following are also in the array if the layer is a stackup layer:
	• Index: integer [the stackup order index]
	• LayerThickness: double [total layer thickness, in meters]
	• EtchFactor: double [This will not appear if the layer has no etch factor defined.]
	• IsIgnored: "true"|"false"
	• NumberOfSublayers: 1 [This is always 1.]
	• Material0: material name
	• FillMaterial0: material name
	• Thickness0: expression with units
	• LowerElevation0: expression with units

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	• If roughness is defined, some or all of the following appear:
	  • Roughness0Type: Groiss | Huray
	  • Roughness0: value and units or surface ratio value for Huray
	  • BottomRoughness0 Type: Groiss | Huray
	  • BottomRoughness0: value and units or surface ratio value for Huray
	  • SideRoughness0 Type: Groiss | Huray
	  • SideRoughness0: value and units or surface ratio value for Huray
	For example:
	`['Type: signal', 'TopBottomAssociation: Neither', 'Color: 32512d', 'IsVisible: true', ' IsVisibleShape: true', ' IsVisiblePath: true', ' IsVisiblePad: true', ' IsVisibleHole: true', ' IsVisibleComponent: true', 'IsLocked: false', 'LayerId: 7', 'Index: 0', 'LayerThickness: 0', 'IsIgnored: false', 'NumberOfSublayers: 1', 'Material0: copper', 'FillMaterial0: FR4_epoxy', 'Thickness0: 0mil', 'Lower-Elevation0: 62mil', 'Roughness0 Type: Huray', 'Roughness0: 0mil, 2.9', 'Bot-tomRoughness0 Type: Huray', 'BottomRoughness0: 0mil, 2.9', 'SideRoughness0 Type: Huray', 'SideRoughness0: 0mil, 2.9']`

Python Syntax	GetLayerInfo(<layer_name>)
Python Example	`oModule = oDesign.GetModule("Footprint")` `oModule.GetLayerInfo ("TopLayer")`

GetProperties (Footprint Editor)

Use: Gets a list of all the properties belonging to a specific **PropServer** and **PropTab**. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: GetProperties(<PropTab>, <PropServer>)

Return Value: Variant array of strings – the names of the properties belonging to the prop server.

GetStackupLayerNames (Footprint Editor)

Use: Informational.

Command: None.

Syntax: GetStackupLayerNames

Return Value: array of strings which are the names of all layers in the layout, blackbox, or footprint.

Parameters: None.

Intersect (Footprint Editor)

Use: Causes Boolean intersecting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: Intersect Array ("NAME: *primitives*",

<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any

…) // etc

Move (Footprint Editor)

Use: Moves an object in the footprint editor

Command: None

Syntax: Move

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PageSetup (Footprint Editor)

Use: Specifies page setup for printing.

Command: **File**>**Page Setup**

Syntax: PageSetup <ArgArray>

Return Value: None.

Parameters: <Margins>: Page margins in implicit units of inches.

<Border>: Integer value indicating to draw border (1) or not to draw (0).

<DesignVars>: Integer value indicating to draw design vars (1) or not to draw (0).

RemoveLayer (Footprint Editor)

Use: Removes a layer or stackup layer.

Command: Remove Layer from a layout or footprint definition

Syntax: RemoveLayer (<LayerName>)

Return Value: None.

Parameters: <LayerName>

Type: <String>

Note As with other Layout scripting interface commands that modifiy the layout, this command is not intended for use within scripts that define footprints. The command behavior from within such a script is undefined and may be unexpected. Use the LayoutHost scripting interface commands within scripts that define footprints.

RemovePort (Footprint Editor)

Selected pins are changed to vias. Selected edge ports are removed.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Not Available**		
Parameters	Name	Type	Description
	<Name>	String	"NAME:elements"
	<ObjectName>	String	Object to be removed; multiple objects can be listed; with each object as a separate string.
Return Value	None		

Python Syntax	RemovePort(["NAME:elements", <*ObjectName*> ...])
Python Example	`oEditor.RemovePort(["NAME:elements","via_195", "Port1"])`

Rotate [Footprint Editor]

Use: Rotates the selected object

Command: None

Syntax: Rotate

Return Value: None

Example:

```
 oDefinitionEditor.Rotate ["NAME:elements", "poly118"],
[0.000469978969405443, 0.000742770842193945]
```

Save (Footprint Editor)

Use: Saves changes during editing of symbols and footprints

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: Save

Return Value: None

Parameters: None

Example:

```
oDefinitionEditor.Save
```

SetActiveDefinitionEditor (Footprint Editor)

Use: Selects the symbol or footprint editor.

Command: None

Syntax: SetActiveDefinitionEditor < EditorName >,<Name>

Return Value: None

Parameters: <EditorName>

Type: <string>

Possible Values: SymbolEditor; FootprintEditor

<Name>

Type: <string>

Name of symbol or footprint to be modified

Example:

```
oProject.SetActiveDefinitionEditor("SymbolEditor", "tqtrx_cap")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetPropertyValue (Footprint Editor)

Use: Sets the value of one property. This is not supported for properties of the following types: **ButtonProp, PointProp,** and **VPointProp**. Only the **ChangeProperty** command can be used to modify these properties. This can be executed by the **oProject**, **oDesign**, or **oEditor** objects.

Command: None

Syntax: SetPropertyValue <PropTab>, <PropServer>, <PropName>, <PropValue>

Return Value: None

Parameters: `<PropValue>`

Type: String

Contains the value to set the property. The formatting is different depending on what type of property is being edited. Use **GetPropertyValue** for the desired property to see the expected format.

Example:

```
oEditor.SetPropertyValue
"BaseElementTab","rect_1",
"LineWidth", "3mm"
```

Subtract (Footprint Editor)

Use: Causes boolean subtracting of one or more *primitive* (polygons, rectangles, lines, or circles) object(s) from another one.

Subtract Array ("NAME: *primitives*",

<object_name>, // Primitive to subtract from

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<object_name>, // 1nd primitive to subtract, if any

…) // etc

Example:

```
oEditor.Intersect Subtract ("NAME:primitives", "circle_0", "rect_2")
```

ToggleViaPin (Footprint Editor)

Use: Selected pins are changed to vias. Selected vias are changed to pins.

Syntax: ToggleViaPin <NAME:elements", <object_name> ... // objects to be toggled

Return Value: None.

Parameters: <object_name>

Type: <String>

Example: `oEditor.ToggleViaPin ["NAME:elements", "via_195"]`

Unite (Footprint Editor)

Use: Causes Boolean uniting of 2 or more *primitive* (polygons, rectangles, lines, or circles) objects.

Syntax: Unite Array ("NAME: *primitives*",

<object_name>, // 1st primitive name

<object_name>, // 2nd primitive, if any

…) // etc

Example:

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oEditor.Unite ["NAME:primitives", "circle_0", "rect_2"]
```

ZoomToFit (Footprint Editor)

Sets the current zoom to fit the contents of the currently visible page.

UI Access	Click the **Fit All** button on the View ribbon or right click in the footprint window and select **View** > **Fit All**
Parameters	None.
Return Value	None

Python Syntax	ZoomToFit()

Python Example	`oProject = oDesktop.SetActiveProject("Project2")`
	`oDefinitionEditor = oProject.SetActiveDefinitionEditor("Foot-printEditor","Distributed Footprints:APC No Ground MCL Footprint 06 Line")`
	`oDefinitionEditor.ZoomToFit()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

27 - Core Global Script Context Commands

To run these commands:

```
import CoreGlobalScriptContextFunctions

CoreGlobalScriptContextFunctions.[CommandName]
```

The following are general script commands recognized by the **CoreGlobalScriptContextFunctions** object:

- AddErrorMessage
- AddFatalMessage
- AddInfoMessage
- AddWarningMessage
- LogDebug
- LogError

AddErrorMessage

Adds an error message to the **Message Manager** window. AddErrorMessageis a function of CoreGlobalScriptContextFunctions.

UI Access	N/A
Parameters	Name: *<message>* — Type: String — Description: Error message.
Return Value	None.

Python Syntax	AddErrorMessage(*<message>*)
Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.AddErrorMessage('My error message.')`

AddFatalMessage

Adds a fatal error message to the **Message Manager** window. AddFatalMessageis a function of CoreGlobalScriptContextFunctions.

UI Access	N/A
Parameters	Name — Type — Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*message*>	String	Error message.
Return Value	None.		

Python Syntax	AddFatalMessage(<*message*>)
Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.AddFatalMessage('My fatal error message.')`

AddInfoMessage

Adds an informational message to the **Message Manager** window. AddInfoMessageis a function of CoreGlobalScriptContextFunctions.

UI Access	N/A		
Parameters	Name	Type	Description
	<*message*>	String	Informational message.
Return Value	None.		

Python Syntax	AddInfoMessage(<*message*>)
Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.AddInfoMessage('My info.')`

AddWarningMessage

Adds a warning message to the **Message Manager** window. AddWarningMessageis a function of CoreGlobalScriptContextFunctions.

UI Access	N/A		
Parameters	Name	Type	Description
	<*message*>	String	Warning message.
Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	AddWarningMessge(<*message*>)
Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.AddWarningMessage('My warning.')`

LogDebug

Adds a debug line to the log specified at **Tools** > **Debug Logging**. LogDebugis a function of CoreGlobalScriptContextFunctions.

UI Access	N/A
Parameters	Name | Type | Description <*message*> | String | Debug message.
Return Value	None.

Python Syntax	LogDebug(<*message*>)
Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.LogDebug('My debug message.')`

LogError

Adds an error line to the log specified at **Tools** > **Debug Logging**. LogErroris a function of CoreGlobalScriptContextFunctions.

UI Access	N/A
Parameters	Name | Type | Description <*error*> | String | Error to log.
Return Value	None.

Python Syntax	LogError(<*error*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`import CoreGlobalScriptContextFunctions` `CoreGlobalScriptContextFunctions.LogError('My error.')`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

28 - Example Scripts

This section contains IronPython example scripts.

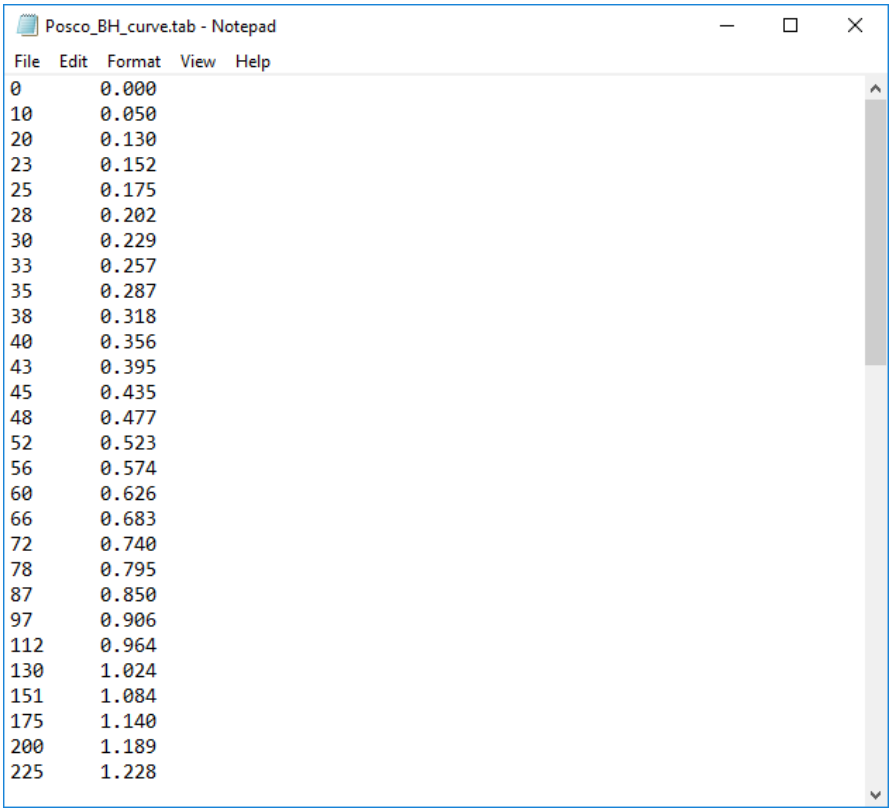
IronPython Example Scripts

IronPython Examples:

- BH Coordinates Python Script
- Equation Based Curve Python Script

BH Coordinates Python Script

This sample Python script adds a material ("Posco 35PN250") with BH coordinates from a specified file (Posco_BH_curve.tab):



To recreate this tab file, paste the text below the script into a text editor and save as Posco_BH_curve.tab.

The script itself includes comment lines, which are preceded by # and offer explanations for the subsequent line(s).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Script Contents

```
# specify path to the file with BH coordinates
path_to_file = r"D:\Posco_BH_curve.tab"
# specify name of the material
material_name = "Posco 35PN250"
oProject = oDesktop.GetActiveProject()
oDefinitionManager = oProject.GetDefinitionManager()
""" create list with B and H points to pass to AddMaterial command
bh_coordinates list is a three dimensional array: 1D: name, 2D:
Coordinate list, 3D: BH point"""
bh_coordinates = ["NAME:BHCoordinates", ["NAME:DimUnits", "", ""]]
with open(path_to_file) as input_file:
  for line in input_file:
    h, b = line.split()

    bh_coordinates.append(["NAME:Coordinate", [
                "NAME:CoordPoint",
                float(h),
                float(b)
              ]
            ])
# create a new magnetic material with BH curve
oDefinitionManager.AddMaterial(
  [
    "NAME:" + material_name,
    "CoordinateSystemType:=", "Cartesian",
    "BulkOrSurfaceType:=" , 1,
    [
      "NAME:PhysicsTypes",
      "set:=" , ["Electromagnetic"]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
      "NAME:permeability",
      "property_type:=" , "nonlinear",
      "BTypeForSingleCurve:=" , "normal",
      "HUnit:=" , "A_per_meter", # unit can be specified as variable
      "BUnit:=" , "tesla", # unit can be specified as variable
      "IsTemperatureDependent:=", False,
      bh_coordinates,
      [
        "NAME:Temperatures"
      ]
    ],
    "conductivity:=" , "1818181.82",
    [
      "NAME:magnetic_coercivity",
      "property_type:=" , "VectorProperty",
      "Magnitude:=" , "0A_per_meter",
      "DirComp1:=" , "1",
      "DirComp2:=" , "0",
      "DirComp3:=" , "0"
    ]
  ])
```

Posco_BH_Curve.tab Contents

```
0       0.000
10      0.050
20      0.130
23      0.152
25      0.175
28      0.202
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
30      0.229
33      0.257
35      0.287
38      0.318
40      0.356
43      0.395
45      0.435
48      0.477
52      0.523
56      0.574
60      0.626
66      0.683
72      0.740
78      0.795
87      0.850
97      0.906
112     0.964
130     1.024
151     1.084
175     1.140
200     1.189
225     1.228
252     1.258
282     1.283
317     1.304
357     1.324
402     1.343
450     1.360
500     1.375
550     1.387
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
602     1.398
657     1.407
717     1.417
784     1.426
867     1.437
974     1.448
1117    1.461
1299    1.478
1515    1.495
1752    1.513
2000    1.529
2253    1.543
2521    1.557
2820    1.570
3167    1.584
3570    1.600
4021    1.617
4503    1.634
5000    1.652
5503    1.669
6021    1.685
6570    1.701
7167    1.717
7839    1.733
8667    1.749
9745    1.769
11167   1.793
12992   1.820
15146   1.851
17518   1.882
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
20000   1.909
22552   1.931
25417   1.948
28906   1.961
33333   1.971
38932   1.981
```

Equation Based Curve Python Script

This sample Python script creates an equation based curve that produces a helix.

```
from math import pi, sin, cos

oProject = oDesktop.GetActiveProject()
oDesign = oProject.GetActiveDesign()
oEditor = oDesign.SetActiveEditor("3D Modeler")

Start_t = 0
End_t = pi*2

Npoint = 128
Nsection = Npoint-1

d_t = (End_t-Start_t)/Nsection

for n in range(1,Nsection):

  P1 = Start_t+d_t*(n-1)
  P2 = P1+d_t
  X_t1 = cos(P1*6)
  Y_t1 = sin(P1*6)
  Z_t1 = P1
  X_t2 = cos(P2*6)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  Y_t2 = sin(P2*6)
  Z_t2 = P2


oEditor.CreatePolyline(
  [
    "NAME:PolylineParameters",
    "IsPolylineCovered:=" , True,
    "IsPolylineClosed:=" , False,
    [
    "NAME:PolylinePoints",
    [
    "NAME:PLPoint",
    "X:=" , '1mm*' + str(X_t1),
    "Y:=" , '1mm*' + str(Y_t1),
    "Z:=" , '1mm*' + str(Z_t1)
    ],
    [
    "NAME:PLPoint",
    "X:=" , '1mm*' + str(X_t2),
    "Y:=" , '1mm*' + str(Y_t2),
    "Z:=" , '1mm*' + str(Z_t2)
    ]
    ],
    [
    "NAME:PolylineSegments",
    [
    "NAME:PLSegment",
    "SegmentType:=" , "Line",
    "StartIndex:=" , 0,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NoOfPoints:=" , 2
    ]
    ],
    [
    "NAME:PolylineXSection",
    "XSectionType:=" , "None",
    "XSectionOrient:=" , "Auto",
    "XSectionWidth:=" , "0mm",
    "XSectionTopWidth:=" , "0mm",
    "XSectionHeight:=" , "0mm",
    "XSectionNumSegments:=" , "0",
    "XSectionBendType:=" , "Corner"
    ]
  ],
  [
    "NAME:Attributes",
    "Name:=" , "Polyline"+str(n),
    "Flags:=" , "",
    "Color:=" , "(132 132 193)",
    "Transparency:=" , 0,
    "PartCoordinateSystem:=", "Global",
    "UDMId:=" , "",
    "MaterialValue:=" , "\"vacuum\"",
    "SurfaceMaterialValue:=", "\"\"",
    "SolveInside:=" , True,
    "IsMaterialEditable:=" , True,
    "UseMaterialAppearance:=", False,
    "IsLightweight:=" , False
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Index

#

2.5D Via Commands 23-1

3D Modeler editor commands 8-1

AlignFaces 8-96

AssignMaterial 8-97

AssignSurfaceMaterial 8-188

Chamfer 8-98

ChangeProperty 6-10, 8-191, 13-8, 26-24

CleanUpModel 8-100

CloseAllWindows 8-195

Connect 8-100

Copy 8-80

CoverLines 8-101

CoverSurfaces 8-102

Create3DComponent 8-6

CreateBondwire 8-11

CreateBox 8-14

CreateCircle 8-16

CreateCone 8-18

CreateCutplane 8-20

CreateCylinder 8-22

CreateEllipse 8-24

CreateEntityList 8-102, 8-109

CreateEquationCurve 8-26

CreateEquationSurface 8-29

CreateFaceCS 8-104, 8-110

CreateGroup 8-107

CreateHelix 8-31

CreateObjectFromdges 8-114

CreateObjectFromFace 8-116

CreateObjectFromFaces 8-117

CreatePoint 8-33

CreatePolyline 8-34

CreateRectangle 8-37

CreateRegion 8-40

CreateRegularPolygon 8-42

CreateRegularPolyhedron 8-44

CreateRelativeCS 8-118

CreateSphere 8-46

CreateSpiral 8-48

CreateTorus 8-49

CreateUserDefinedPart 8-51

Delete 8-195

DeleteEmptyGroups 8-120

DeleteLastOperation 8-121

DeleteOperation 8-121

DeletePolylinePoint 8-81

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DetachEdges 8-123

DetachFaces 8-124

DuplicateAlongLine 8-81

DuplicateAroundAxi 8-83

DuplicateMirror 8-85

EditEntityList 8-125, 8-130

EditFaceCS 8-126

EditObjectCS 8-131

EditPolyline 8-60

EditRelativeCS 8-136

Export 8-137

ExportModelImagetoFile 8-139, 10-55

Fillet 8-143

FlattenGroup 8-144

GenerateAllUserDefinedModels 8-196

GenerateHistory 8-145

GenerateUserDefinedModel 8-197

Get3DComponentParameters 8-65

Get3DComponentPartNames 8-66

GetBodyNamesByPosition 8-199

GetEdgeByPosition 8-205

GetEdgeIDsfromFace 8-207

GetEdgeIDsfromObject 8-207

GetEdgeLength 8-208

GetEntityListIDByName 8-208, 8-209

GetFaceArea 8-209

GetFaceByPosition 8-211

GetFaceCenter 8-210

GetFaceIDs 8-212

GetFaceIDsOfSheet 8-212

GetModelBoundingBox 8-214

GetPartsForUserDefinedModel 8-222

GetPoints 8-222

GetProperties 2-78, 6-40, 7-83

GetPropertyValue 2-79, 6-41, 7-84, 8-223, 10-75, 25-73

GetPropEvaluatedValue 6-37, 8-223

GetPropSIValue 6-39, 8-226

GetRelativeCoordinateSystems 8-228

GetUser Position 8-230

GetVertexIDFromNameForFirstOperation 8-230

GetVertexIDsFromEdge 8-231

GetVertexIDsFromFace 8-231

GetVertexIDsFromObject 8-232

GetVertexPosition 8-232

GetWireBodyNames 8-233

HealObject 8-146

Import 8-148

ImportDXF 8-151, 26-110

ImportFromClipboard 8-153

ImportGDSII 8-154

Imprint 8-156

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImprintProjection 8-157
Insert3DComponent 8-66
InsertComponent 8-68
Intersect 8-159
Mirror 8-87
Move 8-89
MoveCSToEnd 8-160
MoveEntityToGroup 8-160
MoveFaces 8-161
OffsetFaces 8-90
PageSetup 8-233
ProjectSheet 8-163
RemoveBadEdges 8-235
RemoveBadFaces 8-235
RemoveBadVertices 8-236
RenamePart 8-236
Replacewith3DComponent 8-164
Rotate 8-92
Scale 8-93
Section 8-166
SeparateBody 8-168
SetModelUnits 8-168
SetPropValue Modeler 8-237
SetTopDownViewDirectionForActiveView 8-242
SetTopDownViewDirectionForAllViews 8-242
SetWCS 8-169
Simplify 8-171
Split 8-173
Stitch 8-175
Subtract 8-176
SweepAlongPath 8-71
SweepAlongVector 8-72
SweepAroundAxis 8-74
SweepFacesAlongNormal 8-75, 8-75, 8-177, 8-177
SweepFacesAlongNormalWithAttributes 8-77
ThickenSheet 8-178
UncoverFaces 8-179
Ungroup 8-182
Unite 8-180
UpdateComponentDefinition 8-78
UpdatePriorityList 8-243
UpgradeVersion 8-244
Validate3DComponent 8-245
WrapSheet 8-182
WriteHistoryTreeLayoutForTest 8-245
3D Modeler editor object commands
GetNumObjects 8-215
GetObjectIDByName 8-215
GetObjectName 8-216
GetObjectNameByEdgeID 8-216
GetObjectNameByFaceID 8-217
GetObjectNameByID 8-217
GetObjectNameByVertexID 8-218

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetObjectShapeType 8-219

GetObjectVolume 8-221

GetObjPath 8-221

A

Activate 24-49

Activation and Deactivation Methods 24-49

Add 16-1, 16-7, 16-10, 16-16, 21-2, 22-2, 25-14, 25-20, 25-131, 25-160, 25-184, 25-209

AddAddSolverOnDemandModel 25-96

AddAllEyeMeasurements 10-6

AddAntennaOverlay 15-10

AddCableToBundle 8-248

AddCartesianLimitLine 10-6

AddCartesianLimitLineFromCurve 10-7

AddCartesianLimitLineFromEquation 10-9

AddCartesianXMarker 10-10

AddCartesianYMarker 10-11

AddCartesianYMarkerToStack 10-11

AddDataset 6-4, 7-7, 25-6

AddDefinitionFromBlock 25-8

AddDefinitionFromLibFile 8-186

AddDeltaMarker 10-12

AddDiffPair 19-3

AddDynamicLink 7-8

AddHole 24-66

AddLevel 26-131

AddMarker 10-13

AddMarkerToPlot 13-4

AddMaterial 6-6, 25-10, 25-41

AddMenuProp 2-48

AddMenuProp2 2-49

AddMessage 4-6

AddModelingProperties 7-10

AddNote 10-14

AddNPortData 25-92

AddPoint 24-64

AddPortsToAllNets 24-159, 24-159, 26-12, 26-13

AddProp 2-51

AddProp2 2-54

AddRadFieldSourceGroup 15-12

AddRefPort 22-3, 24-160, 26-14

AddRefPortUsingEdges 22-7

AddSweep 21-19

AddTraceCharacteristics 10-15

AddTraces 10-17

AlignFaces 8-96

AlignHorizontal 26-131

AlignObjects 24-182, 26-17

AlignPorts 24-163, 26-19

AlignVertical 26-132

Analysis module commands

Analyze 7-10, 21-21

AnalyzeAll 7-11

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AnalyzeAllNominal 7-11

CopySetup 11-7, 11-30

Analyze

Analysis module command 7-10, 21-21

AnalyzeSweep 21-22

Ansoft Application Object commands 3-1

GetAppDesktop 3-2

ApplyMeshOps 7-12, 7-67

ApplyReportTemplate 10-18

Area 24-70

AreMaterialPropertiesEqual 25-28

AreSurfaceMaterialPropertiesEqual 25-29

AreThereSimulationsRunning 4-7

AssignCircuitRefPort 24-164, 26-20

AssignMaterial 8-97

AssignRefPort 26-21

AssignSurfaceMaterial 8-188

B

BBoxLL 24-68

BBoxUR 24-68

Boolean Operations on Primitives 24-74

Boundary/Excitation module commands

CreateNportCircuitElements 26-55

BreakUDMConnection 8-190

BringToFront 25-218, 26-133

C

Cable Modeling Commands 8-247

Cable Modeling commands

AddCableToBundle 8-248

CreateCableBundle 8-249

CreateCableHarness 8-250

CreateClockSource 8-252

CreatePWLSource 8-253

CreateStraightWireCable 8-255, 8-256

ExportCableLibrary 8-257

ImportCableLibrary 8-258

RemoveCables 8-259

UpdateCableHarness 8-259

CalcOp 13-5, 14-3

CalcRead (deprecated) 14-3

CalcStack 13-6, 14-5

CalculatorRead 13-6, 14-4

CalculatorWrite 13-7, 14-5

Cascade 19-4

Chamfer 8-98

ChangeGeomSettings 13-11, 14-6

ChangeOptions 24-184, 26-184

ChangeProperty 2-41, 6-10, 7-12, 8-191, 10-19, 13-8, 24-189, 25-30, 26-24, 26-134

CircleIntersectsPolygon 24-71

ClcEval 13-12, 14-7

ClcMaterial 13-12, 14-7

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClcMaterialValue 14-8

CleanUpModel 8-100

ClearAllMarkers 10-26

ClearAllNamedExpr 13-13, 14-9

ClearAllTraceCharacteristics 10-26

ClearDiffPairs 19-5

ClearLayerMappings 24-130, 26-27

ClearMessages 4-7, 6-13

ClearRefPort 24-165, 26-28

ClearRelative 24-81, 26-29

ClipPlane 24-195

Clone 19-6

CloneMaterial 6-14, 25-40

CloneReportsFromDatasetSolution 10-27

Close 6-15, 19-7

CloseAllWindows 4-11, 8-195

CloseEditor 26-144, 26-189

CloseProject 4-12

CloseProjectNoForce 4-12

ClosestPointOnLine 24-61

Combine 19-8

CompInstance Functions 56

CompInstance Script Commands (multiple)

Component Manager Script Commands 25-89

Connect 8-100, 24-43, 26-29

conventions

scripting help 1-1

ConvertPlaneToTrace 22-7, 26-30

ConvertPrimitives 23-1, 24-74

ConvertToDynamic 25-166

ConvertToParametric 25-166

ConvertTraceToPlane 22-8, 26-31

Coordinate System Methods 24-78

Copy 8-80, 24-44, 26-31, 26-144

CopyDesign 6-15

CopyItemCommand 7-19

CopyNamedExprToStack 13-13, 14-9

CopyPlotSettings 10-28

CopyRadFieldSetup 15-2

CopyReportDefinition 10-28

CopyReportsData 10-29

Copyright and Trademark Information 2

CopySetup

Analysis module command 11-7, 11-30

Optimetrics module command 11-7, 11-30

CopyTraceDefinitions 10-29

CopyTracesData 10-30

Core Global Script Context Commands 27-1

Count 4-13

CoupleEdgePorts 22-9

CoverLines 8-101

CoverSurfaces 8-102

CPython 1-44

Create Primitives 24-3

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Create Voids in Primitives 24-18

Create3DComponent 8-6

Create3DStructure 24-82, 26-32

CreateArc 26-145

CreateBondwire 8-11

CreateBox 8-14

CreateCableBundle 8-249

CreateCableHarness 8-250

CreateCircle 8-16, 24-6, 26-36, 26-146, 26-189

CreateCircuitPort 24-157, 24-166, 26-7, 26-39

CreateClockSource 8-252

CreateComponent 24-26, 24-26, 26-41, 26-41

CreateCone 8-18

CreateCS 24-79, 24-79, 26-33, 26-33

CreateCutplane 8-20

CreateCylinder 8-22

CreateEllipse 8-24

CreateEntityList 8-102, 8-109

CreateEquationCurve 8-26

CreateEquationSurface 8-29

CreateFaceCS 8-104, 8-110

CreateFieldPlot 13-14

CreateGroundPortComponent 22-9, 26-45

CreateGroup 8-107

CreateGroupSelected 24-84, 26-46

CreateHelix 8-31

CreateHole 24-28, 26-47

CreateImage 26-146

CreateInterfaceGround 22-10, 26-48

CreateInterfacePort 22-10, 26-49

CreateInterfacePortComponent 22-10, 26-49

CreateLine 24-7, 26-49, 26-147, 26-193

CreateLineFromPolygon 24-52, 26-51

CreateLineVoid 24-19, 26-51, 26-195

CreateNetClass 24-142, 26-57

CreateNportCircuitElements 26-55

CreateObjectFromEdges 8-114

CreateObjectFromFace 8-116

CreateObjectFromFaces 8-117

CreateOutputVariable 9-1

CreatePin 26-148

CreatePinGroup 24-34, 26-59

CreatePinGroupPort 24-34, 26-59

CreatePoint 8-33

CreatePolygon 24-12, 26-61, 26-148

CreatePolygonVoid 24-21, 26-63, 26-195

CreatePolyline 8-34

CreatePortInstancePorts 24-171, 26-65

CreatePortsOnComponents 24-171, 26-65

CreatePWLSource 8-253

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateRectangle 8-37, 24-15, 26-66, 26-149, 26-199

CreateRectangleVoid 24-22, 26-67

CreateRegion 8-40

CreateRegularPolygon 8-42

CreateRegularPolyhedron 8-44

CreateRelativeCS 8-118

CreateReport 7-19, 10-31

CreateReportFromTemplate 10-43

CreateReportofAllQuantities 10-44

CreateSphere 8-46

CreateSpiral 8-48

CreateText 24-16, 26-69, 26-149, 26-199

CreateTorus 8-49

CreateTrace 24-35, 26-70

CreateUserDefinedPart 8-51

CreateUserDefinedSolution 18-1

CreateVia 24-38, 26-73, 26-199

Cross 24-60

Cut 26-150

CutDesign 6-16

CutOutSubDesign 24-196, 26-74

D

dataset commands

AddDataset 6-4, 7-7, 25-6

DeleteDataset 6-17, 7-32, 25-40

EditDataset 6-18, 7-37, 25-54

ExportDataset 6-27, 7-52, 25-68

ImportDataset 6-45, 7-88, 25-74

DeactivateOpen 24-50, 26-77

DeactivateShort 24-50, 26-78

DecoupleEdgePorts 22-11

Deembed 19-9

DeembedBack 19-10

DeembedFront 19-11

DefeatureObjects 24-200, 26-78

Definition Editor Script Commands 26-1

Definition Manager commands

AddDefinitionFromBlock 25-8

AddDefinitionFromLibFile 8-186

GetProjectMaterialNames 25-72

UpdateDefFromBlock 25-83, 25-85

Definition Manager Script Commands 25-1

Delete 8-195, 21-23, 22-11, 22-11, 23-2, 24-44, 24-44, 24-51, 24-51, 26-79, 26-79, 26-150

Delete [Interface Port] 22-11

DeleteAllReports 10-45

DeleteDataset 6-17, 7-32, 25-40

DeleteDesign 6-17

DeleteDesignInstance 7-33, 7-33

DeleteEmptyGroups 8-120

DeleteFarFieldSetup 15-2

DeleteFieldPlot 13-25

DeleteFieldVariation 7-33

DeleteFullVariation 7-34

DeleteImportData 12-1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeleteLastOperation 8-121

DeleteMarker 10-45

DeleteMarker[Fields Reporter] 13-25

DeleteNamedExpr 13-26, 14-10

DeleteNearFieldSetup 15-3

DeleteOperation 8-121

DeleteOutputVariable 7-35, 9-3

DeletePinGroup 24-42, 26-80

DeletePolylinePoint 8-81

DeleteProbePortAndVia 22-11

DeleteProject 4-14

DeleteReport 10-46

DeleteSetup 15-4

DeleteSetups

Optimetrics module command 11-8, 11-31

DeleteSolutionVariation 7-35, 12-2

DeleteSource [Interface Source] 22-12

Deletetraces 10-46

DeleteUneditablePlot 13-26

DeleteUserDefinedSolutions 18-2

DelNetClass 24-143, 26-79

Design object commands 10-64, 10-65, 10-66, 10-67, 10-68, 10-80

AddCartesianLimitLine 10-6

AddCartesianXMarker 10-10

AddCartesianYMarker 10-11

AddDeltaMarker 10-12

AddDynamicLink 7-8

AddMarker 10-13

AddMarkerToPlot 13-4

AddNote 10-14

AddTraceCharacteristics 10-15

AddTraces 10-17

ApplyMeshOps 7-12, 7-67

CalculatorRead 13-6, 14-4

ClearAllMarkers 10-26

ClearAllTraceCharacteristics 10-26

CloseAllWindows 4-11

CopyReportDefinition 10-28

CreateOutputVariable 9-1

CreateReportFromTemplate 10-43

DeleteAllReports 10-45

DeleteOutputVariable 7-35, 9-3

DeleteReport 10-46

DeleteTraces 10-46

DoesOutputVariableExist 9-3

Edit Layout 7-41

EditMarker 10-49

EditOutputVariable 7-43, 9-4

ExportConvergence 7-52

ExportMeshStats 7-57

ExportModelMeshToFile 8-142, 10-58

ExportPlot3DToFile 10-59

ExportPlotImageToFile 13-41

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportPlotImageWithViewToFile 13-42

ExportProfile 7-63

ExportReport 7-64, 10-60

ExportToFile 10-62

Fast Transformation 7-109, 7-110

GeometryCheckAndAutofix 8-197, 26-95

GetDesignValidationInfo 7-71

GetDisplayType 10-71

GetLibraryDirectory 4-27

GetMatchedObjectName 8-213

GetModelUnits 8-214

GetModule 7-75, 7-76, 7-77

GetName 7-78, 10-74, 11-18, 11-45

GetObjectsByMaterial 8-219

GetObjectsInGroup 8-220

GetObjPath 7-80, 10-74, 11-19, 11-46

GetOutputVariableValue 7-81, 9-7

GetProjectDirectory 4-31

GetProperties 2-78, 6-40, 7-83

GetPropertyValue 2-79, 6-41, 7-84, 8-223, 10-75, 25-73

GetPropEvaluatedValue 6-37, 8-223

GetPropNames 7-86, 15-6

GetPropSIValue 6-39, 8-226

GetRegistryInt 4-33, 4-68

GetRegistryString 4-33, 4-68

GetSelections 8-228

GetSubGroupsInGroup 8-229

GetTempDirectory 4-37

GetVariables 2-80, 6-44, 7-87

GetVariableValue 2-81, 6-43, 7-88

GetVersion 4-38

ImportIntoReport 10-93

IsFeaturEnabled 4-38

PasteDesign 7-93

PasteReports 10-98

PasteReportsWithLegacyNames 10-98

PasteTracesWithLegacyNames 10-99

Redo 7-96

RenameDesignInstance 7-98

RenameReport 10-100

RenameTraces 10-101

RunToolkit 6-53, 7-100

SavePlotSettingsAsDefault 10-102

SetActiveEditor 7-82, 7-101

SetLibraryDirectory 4-57

SetProjectDirectoryVBCommand> 4-58

SetPropertyValue 2-81, 6-60, 7-113, 25-82

SetTempDirectory 4-61

SetVariableValue 2-83, 6-61, 7-115

Show Layout 7-114

ShowWindow 8-170

Solve 7-115

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateAllReports 10-105

Design Object Script Commands 7-1

Design Object SCripting commands

ExportNetworkData 7-58

Desktop Commands For Registry Values 4-65

Desktop object commands

AddMessage 4-6

AreThereSimulationsRunning 4-7

ClearMessages 4-7

CloseProject 4-12

CloseProjectNoForce 4-12

Count 4-13

DeleteProject 4-14

DownloadJobResults 4-14

EnableAutosave 4-17

ExportOptionsFiles 4-17

GetActiveProject 4-18

GetActiveScheduler 4-18

GetActiveSchedulerInfo 4-19

GetAutosaveEnabled 4-20

GetBuildTimeDateString 4-21

GetChildNames 6-30, 7-68, 15-5

GetChildNames Modeler 8-200

GetChildTypes 6-31, 7-69, 8-204, 15-6

GetChildTypes Optimetrics 11-18, 11-44

GetCustomMenuSet 4-21

GetDesigns 6-33

GetDesktopConfiguration 4-24

GetDistributedAnalysisMachines 4-24

GetDistributedAnalysisMachinesForDesignType 4-25

GetMonitorData 4-29

GetProjectList 4-31

GetProjects 4-28, 4-32

GetPropNames 6-38, 11-21, 11-47

GetSchematicEnvironment 4-35

KeepDesktopResponsive 4-39

LaunchJobMonitor 4-40

NewProject 4-41

OpenProject 4-42

PageSetup 4-43, 24-222, 26-114, 26-205

PauseRecording 4-44

PauseScript 4-44

Print 4-45

QuitApplication 4-45

RefreshJobMonitor 4-46

ResetLogging 4-47

RestoreWindow 4-48

ResumeRecording 4-49

RunProgram 4-50

RunScript 4-50

SelectScheduler 4-53

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetActiveProject 4-54

SetActiveProjectByPath 4-55

SetCustomMenuSet 4-55

SetDesktopConfiguration 4-56

SetSchematicEnvironment 4-60

Sleep 4-62

StopSimulations 4-63

SubmitJob 4-64

TileWindows 4-65

Desktop Object Script Commands 4-1

Desktop Scripting Conventions 1-57

DetachEdges 8-123

DetachFaces 8-124

DisableDiffPairs 19-12

Disconnect 24-44, 26-80

DisplayPinNames 26-151

Distance 24-60

DistanceFromLine 24-61

DistributedAnalyzeSetup, Optimetrics module command 11-8, 11-32

DoesNamedExpressionExists 13-27, 14-10

DoesOutputVariableExist 9-3

DoesSupportTraceCharacteristics 10-47

Draw Menu commands 8-4

Duplicate 24-202, 26-81, 26-199

DuplicateAcrossLyrs 24-130, 26-81

DuplicateAlongLine 8-81

DuplicateAroundAxis 8-83

DuplicateMirror 8-85

DynamicMeshOverlays 21-24

E

Edit 7-40, 16-4, 16-9, 16-13, 16-19, 21-25, 21-25, 22-12, 22-12, 23-2, 24-45, 25-42, 25-63, 25-105, 25-143, 25-166, 25-190, 25-218, 26-82

Edit Menu Commands 8-79

Edit3DComponent 8-56

Edit3DComponentDefinition 8-58, 24-202, 26-83

EditAntennaArraySetup 15-45

EditAntennaOverlay 15-13

EditBoxSetup 15-14

EditCartesianXMarker 10-48

EditCartesianYMarker 10-49

EditCircuitExcitations 22-23

EditCircuitPort [Interface Port] 22-23

EditCoSimulationOptions 7-36

EditDataset 6-18, 7-37, 25-54

EditEntityList 8-125, 8-130

EditFaceCS 8-126

EditFarFieldSphereSetup 15-16

EditHorizontalWavePort 22-25

EditImportData 7-39

EditInfiniteArray 7-39

EditInfiniteSphereSetup 15-17

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditLayoutForLayoutComponent 7-41
EditLineSetup 15-19
EditMarker 10-49
EditMaterial 6-20, 25-56
EditNearFieldBoxSetup 15-20
EditNearFieldLineSetup 15-22
EditNearFieldRectangleSetup 15-23
EditNearFieldSphereSetup 15-24
EditNotes 7-42, 7-42
EditObjectCS 8-131
EditOptions 7-43
Editor object commands
SetPropertyValue 2-81, 6-60, 7-113, 25-82
EditOutputVariable 7-43, 9-4
EditPolyline 8-60
EditQ3DSources 12-3
EditRadFieldSourceGroup 15-26
EditRectangleSetup 15-27
EditRelativeCS 8-136
EditSetup
optimization command 11-61
parametric command 11-54
sensitivity command 11-70
statistical command 11-74
EditSolverOnDemandModel 25-117
EditSources 12-6
EditSphereSetup 15-28
EditSurfaceMeshSummaryData 13-27
EditUserDefinedSolution 18-3
EditWithComps 25-218
EMDesignOptions 7-44
EnableAutoSave 4-17
EnableDiffPairs 19-13
EnableSetup 11-9, 11-32
EnterComplex 13-29, 14-11
EnterComplexVector 13-30, 14-11
EnterCoord 13-31, 14-12
EnterEdge 13-31, 14-13
EnterLine 13-32, 14-13
EnterOutputVar 13-33, 14-14
EnterPoint 13-33, 14-14
EnterQty 13-34, 14-15
EnterScalar 13-34, 14-16
EnterScalarFunc 13-35, 14-16
EnterSurf 13-35, 14-17
EnterVector 13-36, 14-17
EnterVectorFunc 13-37, 14-18
EnterVol 13-37, 14-18
EraseMeasurements 24-207, 26-84, 26-200
Example Scripts 28-1
Excitations Commands 22-1
ExecuteScript 2-56
Export 8-137, 25-68, 25-70, 25-71, 25-124, 25-153, 25-173, 25-201, 25-227

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportCableLibrary 8-257

ExportCitiFile 19-13

ExportConvergence 7-52

ExportDataset 6-27, 7-52, 25-68

ExportDOELocalSensitivity 11-41

ExportDOELocalSensitivityCurve 11-42

ExportDOEResponseCurve 11-39

ExportDOEResponseCurveSlices 11-40

ExportDOEResponseSurface 11-40

ExportDXConfigFile, Optimetrics module command 11-10, 11-33

ExportDXF 24-207, 26-86

ExportEigenmodes 12-10

ExportEignemodes 12-10

ExportElementPatternToFile 15-50

ExportEyeMaskViolation 10-50

ExportFieldPlot 13-38

ExportFieldsToFile 15-51

ExportFile 26-151

ExportForHSpice 7-53, 12-11

ExportForHSpice (Layout Editor) 7-54

ExportForSpice (Layout Editor) 7-56

ExportFullWaveSpice 19-15, 25-177

ExportGDSII 24-131, 24-209, 26-87

ExportGerber 24-211, 26-89

ExportImageToFile 10-51

ExportMarkerTable 13-38

ExportMaterial 6-28, 25-69

ExportMatlab 19-19

ExportMesh Stats 7-57

ExportModelMeshToFile 8-142, 10-58

ExportNCDrill 24-212, 26-91

ExportNMFData 12-14

ExportOptimetricsProfile, Optimetrics module command 11-10, 11-34

ExportOptimetricsResults, Optimetrics module command 11-11, 11-34

ExportOptionsFiles 4-17

ExportOutputVariables 9-5

ExportParametersToFile 15-53

ExportParametricResults, Optimetrics module command 11-12, 11-35

ExportPlot3DToFile 10-59

ExportPlotImageToFile 13-41

ExportPlotImageWithViewToFile 13-42

ExportProfile 7-63

ExportRadiationFieldsToFile 15-54

ExportRadiationParametersToFile 15-55

ExportReport 7-64, 10-60

ExportReportDataToFile 10-61

ExportRespSurfaceMinMaxTable 11-13, 11-36

ExportRespSurfaceRefinePoints 11-14, 11-37

ExportRespSurfaceResponsePoints 11-15, 11-38

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportRespSurfaceVerificationPoints 11-15, 11-38

ExportScript 25-71, 25-206

ExportSpreadsheet 19-24

ExportSurfaceMeshSummary 13-43

ExportTableToFile 10-61

ExportToFile 13-44

ExportToHFSS 21-38, 26-93

ExportToQ3D 21-39, 26-93

ExportTouchstone 19-26

ExportTouchstone2 19-28

ExportUniformPointsToFile 10-63

Extract 19-30

F

FFTONReport 12-15

Field Calculator commands

DoesNamedExpressionExists 13-27, 14-10

Field Overlay module commands, GetFieldPlotNames 13-45

field overlay script commands 13-1

Field Overlays module commands

CalcOp 13-5, 14-3

CalcStack 13-6, 14-5

ChangeGeomSettings 13-11, 14-6

ClcMaterial 13-12, 14-7

ClearAllNamedExpr 13-13, 14-9

CopyNamedExprToStack 13-13, 14-9

CreateFieldPlot 13-14

DeleteFieldPlot 13-25

DeleteNamedExpr 13-26, 14-10

DeleteUneditablePlot 13-26

EditSurfaceMeshSummaryData 13-27

EnterComplex 13-29, 14-11

EnterComplexVector 13-30, 14-11

EnterCoord 13-31, 14-12

EnterEdge 13-31, 14-13

EnterLine 13-32, 14-13

EnterOutputVar 13-33, 14-14

EnterPoint 13-33, 14-14

EnterQty 13-34, 14-15

EnterScalar 13-34, 14-16

EnterScalarFunc 13-35, 14-16

EnterSurf 13-35, 14-17

EnterVector 13-36, 14-17

EnterVectorFunc 13-37, 14-18

EnteVol 13-37, 14-18

ExportFieldPlot 13-38

ExportOnGrid 13-39

ExportSurfaceMeshSummary 13-43

GetFieldFolderNames 13-45

GetFieldPlotQuantityName 13-46

GetMeshPlotNames 13-46, 21-40

HidePolarPlot 13-48

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HideRadiatedPlotOverlay 13-49
ModifyFieldPlot 13-50
ReassignFieldPlot 13-52
RenameFieldPlot 13-53
RenamePlotFolder 13-54
SaveFieldsPlots 13-55
SetFieldPlotSettings 13-56
SetPlotFolderSettings 13-58
UpdateAllFieldsPlots 13-62
UpdateQuantityFieldsPlots 13-63
Fields Calculator commands
AddNamedExpr 13-5, 14-2
CalcOp 13-5, 14-3
CalcStack 13-6, 14-5
CalculatorWrite 13-7, 14-5
ChangeGeomSettings 13-11, 14-6
ClcEval 13-12, 14-7
ClcMaterial 13-12, 14-7
ClcMaterialValue 14-8
ClearAllNamedExpr 13-13, 14-9
CopyNamedExprToStack 13-13, 14-9
DeleteNamedExpr 13-26, 14-10
EnterComplex 13-29, 14-11
EnterComplexVector 13-30, 14-11
EnterCoord 13-31, 14-12
EnterEdge 13-31, 14-13
EnterLine 13-32, 14-13
EnterOutputVar 13-33, 14-14
EnterPoint 13-33, 14-14
EnterQty 13-34, 14-15
EnterScalar 13-34, 14-16
EnterScalarFunc 13-35, 14-16
EnterSurf 13-35, 14-17
EnterVector 13-36, 14-17
EnterVectorFunc 13-37, 14-18
EnterVol 13-37, 14-18
ExportOnGrid 14-19
ExportToFile 14-21
Fields Calculator Script Commands 14-1
Fields reporter module commands
DeleteMarker[Fields Reporter] 13-25
ExportMarkerTable 13-38
Fillet 8-143
FilterObjectList 24-54, 26-93
FindObjects 24-53
FindObjectsByPoint 24-55, 26-94
FindObjectsByPolygon 24-55, 26-95
FlattenGroup 8-144
FlipHorizontal 24-48, 26-95, 26-152, 26-201
FlipVertical 24-48, 26-95, 26-152, 26-201
footprint
editor scripts 26-167
manager script commands 25-130

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

G

GenerateAllUserDefinedModels 8-196

GenerateHistory 8-145

GenerateUserDefinedModel 8-197

GenerateVariationData Parametric, parametric command 11-16, 11-42, 11-56

GeometryCheckAndAutofix 8-197, 26-95

Get3DComponentDefinitionNames 8-63

Get3DComponentInstanceNames 8-64

Get3DComponentMaterialNames 8-64

Get3DComponentMaterialProperties 8-65

Get3DComponentParameters 8-65

Get3DComponentPartNames 8-66

GetActiveDesign 6-28

GetActiveProject 4-18

GetActiveScheduler 4-18

GetActiveSchedulerInfo 4-19

GetAdaptiveFreq 12-16

GetAllBoundariesList 22-26

GetAllCategories 10-64

GetAllLayerNames 24-133, 26-97, 26-201

GetAllPortsList 22-26

GetAllQuantities 10-65

GetAllReportNames 10-66

GetAllSolutionNames 21-39

GetApplication 2-56

GetArrayVariables 2-77, 6-29

GetAutoSaveEnabled 4-20

GetAvailableDisplayTypes 10-67

GetAvailableReportTypes 10-67

GetAvailableSolutions 10-68

GetAvailableVariations 12-17

GetBBox 24-56, 26-97

GetBodyNamesByPosition 8-199

GetBoundingCircleCenter 24-70

GetBoundingCircleRadius 24-70

GetBuildDateTimeString 4-21

GetCallback 2-57

GetChangedProperty 2-58

GetChild Object (Report Setup), Report module command 10-69

GetChildNames 6-30, 7-68, 15-5

GetChildNames (Optimetrics), Report module command 10-68

GetChildNames Modeler 8-200

GetChildObject Design 6-30, 7-68

GetChildObject Modeler 8-202

GetChildObject Radiation 15-5

GetChildTypes 6-31, 7-69, 8-204, 15-6

GetChildTypes Optimetrics 11-18, 11-44

GetChildTypes ReportSetup 10-69

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetClosestPoint 24-72

GetClosestPoints 24-72

GetCompInstanceFromRefDes 24-216, 26-97

GetComponentInfo 24-215, 26-98

GetComponentName 56

GetComponentPinInfo 24-217, 26-99

GetComponentPins 24-216, 26-99

GetCSObjects

Coordinate Systems 24-56, 24-83, 26-97

GetCurvePropServerName 10-70

GetDataUnits 10-82

GetDefinitionManager 6-32

GetDependentFiles 6-32

GetDescription 2-58

GetDesign 2-59, 2-59

GetDesignID 7-69

GetDesignName 7-70

GetDesigns 6-33

GetDesignType 7-71

GetDesignValidationInfo 7-71

GetDesignVariableNames 10-82

GetDesignVariableUnits 10-83

GetDesignVariableValue 10-84

GetDesignVariationKey 10-85

GetDisplayType 10-71

GetDistributedAnalysisMachines 4-24

GetDistributedAnalysisMachinesForDesignType 4-25

GetDocumentNames 17-5

GetDynLinkIntrinsicVariables 10-71

GetDynLinkQtyValueState 10-72

GetDynLinkTraces 10-72

GetDynLinkVariableValues 10-73

GetEdgeByPosition 8-205

GetEdgeIDsFromFace 8-207

GetEdgeIDsFromObject 8-207

GetEdgeLength 8-208

GetEditor 2-59, 7-72

GetEditor (Layout Editor) 7-72

GetEditorName [Layout] 24-218, 26-100

GetEntityIDsContainedByNamedSelection 8-208

GetEntityListIDByName 8-209

GetEvaluatedText 2-60

GetExeDir 4-26

GetFaceArea 8-209

GetFaceByPosition 8-211

GetFaceCenter 8-210

GetFaceIDs 8-212

GetFaceIDsOfSheet 8-212

GetFieldFolderNames 13-45

GetFieldPlotNames, Field Overlay module command 13-45

GetFieldPlotQuantityName 13-46

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetFileName 2-60

GetFrequencies 19-31

GetFrequencyCount 19-32

GetGeometryIdsForAllNetLayerCombinations 7-74

GetGeometryIdsForNetLayerCombinations 7-73

GetHidden 2-61

GetHoles 24-67

GetImagDataValues 10-85

GetInstanceID 57

GetInstanceName 57

GetIntersectionType 24-71

GetLayerInfo 24-134, 26-201

GetLibraryDirectory 4-27

GetMatchedObjectName 8-213

GetMaterialList 24-218, 26-101

GetMeshOperations 21-40

GetMeshPlotNames 13-46

GetModelBoundingBox 8-214

GetModelUnits 8-214

GetModule 7-75, 7-76, 7-77

GetMonitorData 4-29

GetName 6-35, 7-78, 7-79, 10-74, 11-18, 11-45, 19-32

GetNames 25-157, 25-202

GetNetClasses 24-143, 26-101

GetNetClassNets 24-144, 26-101

GetNetConnections 24-144, 26-102

GetNumObjects 8-215

GetObjectIDByName 8-215

GetObjectName 8-216

GetObjectNameByEdgeID 8-216

GetObjectNameByFaceID 8-217

GetObjectNameByID 8-217

GetObjectNameByVertexID 8-218

GetObjectsByMaterial 8-219

GetObjectShapeType 8-219

GetObjectsIngroup 8-220

GetObjectVolume 8-221

GetObjPath 6-36, 7-80, 8-221, 10-74, 11-19, 11-46

GetOptimetricsResults, Optimetrics module command 11-20

GetOutputVariableValue 7-81, 9-7

GetParentDesign 58

GetPartsForUserDefinedModel 8-222

GetPath 6-36

GetPerQuantityPrimarySweepValues 10-86

GetPersonalLibDirectory 4-30

GetPoints 8-222, 24-66

GetPolygon 24-56, 26-103

GetPolygonDef 24-56, 26-103

GetPortCount 19-33

GetPortInfo 24-173, 26-103

GetPortNumber 19-34

GetPostProcSettings 19-35

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetProcessID 4-30

GetProgress 2-62, 2-62

GetProjectDirectory 4-31

GetProjectList 4-31

GetProjectMaterialNames 25-72

GetProjects 4-28, 4-32

GetProperties 2-78, 6-40, 7-83, 24-218, 25-157, 26-106, 26-153, 26-204

GetPropertyValue 2-79, 6-41, 7-84, 8-223, 10-75, 24-219, 25-73, 26-106, 26-153

GetPropEvaluatedValue 6-37, 8-223

GetPropHost 2-62, 2-62, 7-83, 7-83, 58

GetPropNames 6-38, 7-86, 10-76, 11-21, 11-47, 15-6

GetPropNames Modeler 8-225

GetPropServerName 59

GetPropServers 2-62

GetPropSIValue 6-39, 8-226

GetPropTabType 2-63

GetPropValue Modeler 8-227

GetPropValue Optimetrics 11-22, 11-47

GetPropValue Project 6-40, 7-86, 15-6

GetPropValue Reporter 10-77

GetQtyExpressionsForSourceTrace 10-77

GetReadOnly 2-63

GetRealDataValues 10-87

GetRegistryInt 4-33, 4-68

GetRegistryString 4-33, 4-68

GetRelativeCoordinateSystems 8-228

GetReportSummaryForRegressionTesting 10-79

GetReportTraceNames 10-78

GetRunStatus 2-64

GetSelections 8-228, 24-219, 24-219, 26-106

GetSetupData 21-40

GetSetupNames 15-4, 15-7

GetSetupNames (Optimetrics), Optimetrics module command 11-16, 11-17, 11-22, 11-43, 11-44, 11-48

GetSetupNamesByType (Optimetrics), Optimetrics module command 11-23, 11-49

GetSolutionContexts 10-79

GetSolutionDataPerVariation 10-80

GetSolutionVariation 19-36

GetSolutionVersionID 12-18

GetSolveRangeInfo 12-18

GetSolverOnDemandData 25-126

GetSolverOnDemandModelList 25-127

GetSourceContexts 12-19

GetSourceData 12-19

GetStackupLayerNames 24-136, 26-204

GetSubGroupsInGroup 8-229

GetSweepNames 10-88

GetSweepUnits 10-89

GetSweepValues 10-89

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSysLibDirectory 4-36

GetTabTypeName 2-64

GetTempDirectory 4-37

GetText 2-65

GetTool 4-71

GetTopDesignList 6-43

GetTopEntryValue 13-47, 14-24

GetUserDefinedSolutionNames 18-4

GetUserDefinedSolutionProperties 18-4

GetUserLibDirectory 4-37

GetUserPosition 8-230

GetValidISolutionList 12-20

GetValue 2-66

GetVariables 2-80, 6-44, 7-87

GetVariableValue 2-81, 6-43, 7-88

GetVariation 19-36

GetVersion 4-38

GetVertexIDFromNameForFirstOperation 8-230

GetVertexIDsFromEdge 8-231

GetVertexIDsFromFace 8-231

GetVertexIDsFromObject 8-232

GetVertexPosition 8-232

GetVisibleLevels 26-153

GetWireBodyNames 8-233

GetX (Layout Editor) 24-59

GetY 24-59

GridSetup 26-154

Group 24-83, 26-106

GroupPlotCurvesByGroupingStrategy 10-93

H

HasArcs 24-67

HasFields 12-20

HasHoles 24-67

HasMatrixData 12-21

HasSameData 19-37

HasSelfIntersections 24-67

Heal (Layout Editor) 24-220, 26-107

HealObject 8-146

HideAntennaParametersOverlay 13-48

HidePolarPlot 13-48

HideRadiatedPlotOverlay 13-49

hierarchy of variables in HFSS 1-50

HighlightNet 24-145, 26-109

I

Icepak boundary condition commands

- ExportDOELocalSensitivity 11-41
- ExportDOELocalSensitivityCurve 11-42
- ExportDOEResponseCurve 11-39
- ExportDOEResponseCurveSlices 11-40
- ExportDOEResponseSurface 11-40

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ImportIDX 4-82
Import (3D Modeler command) 8-148
ImportANF 4-72, 4-72
ImportAutoCAD 4-73
ImportAWRMicrowaveOffice 4-74
ImportCableLibrary 8-258
ImportDataset 6-45, 7-88, 25-74
ImportDXF 8-151, 26-110
ImportEDB 4-75
ImportExport Tool 4-71
ImportExtracta 4-75
ImportFromClipboard 8-153
ImportGDSII 4-76, 4-77, 8-154, 26-113
ImportIDF 4-78, 4-81
ImportIDX 4-82
ImportIntoReport 10-93
ImportIPC 4-84
ImportODB 4-84
ImportOutputVariables 9-8
ImportReportDataIntoReport 10-94
ImportSetup, Optimetrics module command 11-23, 11-49
ImportSolution 12-22
ImportTable 12-23
ImportXFL 4-85
Imprint 8-156
ImprintProjection 8-157
Insert3DComponent 8-66
InsertBoxSetup 15-30
InsertComponent 8-68
InsertDesign 6-46, 6-47, 7-90
InsertDesignWithWorkflow 6-48
InsertFarFieldSphereSetup 15-32
InsertInfiniteSphereSetup 15-33
InsertLineSetup 15-35
InsertNearFieldBoxSetup 15-36
InsertNearFieldLineSetup 15-38
InsertNearFieldRectangleSetup 15-39
InsertNearFieldSphereSetup 15-40
InsertRectangleSetup 15-42
InsertSetup
optimization command 11-64
parametric command 11-28, 11-53, 11-56, 11-60
statistical command 11-75
InsertSphereSetup 15-43
Intersect 8-159, 24-73, 24-73, 24-76, 24-76, 26-113, 26-113, 26-204
IronPython 1-2
IsArc 24-60, 24-60, 24-69, 24-69
IsBox 24-69
IsCircle 24-69
IsClosed 24-65
IsConvex 24-69
IsDataComplex 10-90
IsEqual 24-60
IsFeatureEnabled 4-38

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

IsFieldAvailableAt 12-24

IsParametric 24-69

IsPerQuantityPrimarySweep 10-91

IsPoint 24-69

IsSegment 24-69

IsUsed 25-174

IsValueConstant 2-66

K

KeepDesktopResponsive 4-39

L

LaunchJobMonitor 4-40

Layers Methods 24-116

layout

script interrogation methods 24-51

Layout Interrogation 24-51

Layout Scripting 1-59, 24-1, 24-88, 24-180

Layout3DMeshOverlay 21-41

LayoutMeshOverlay 21-41

ListMatchingVariations 12-25

ListValuesOfVariable 12-25

ListVariations 21-41

LoadNamedExpressions 13-49, 14-24

LoadSolution 19-39

M

Mag (Layout Editor) 24-60

material commands

AddDefinitionFromBlock 25-8

AddDefinitionFromLibFile 8-186

AddMaterial 6-6, 25-10, 25-41

AreMaterialPropertiesEqual 25-28

AreSurfaceMaterialPropertiesEqual 25-29

CloneMaterial 6-14, 25-40

EditMaterial 6-20, 25-56

ExportMaterial 6-28, 25-69

RemoveMaterial 6-51, 25-78

UpdateDefFromBlock 25-83

UpdateDefFromBlockEx 25-85

Mirror 8-87

MirrorX 24-66

Model Manager Script Commands 25-159

Modeler Menu Commands 8-94

ModifyFieldPlot 13-50

ModifyNetClass 24-147, 26-114

module script commands

field overlay 13-1

radiation 15-2

ModuleHasMesh 12-21

modules in HFSS scripting 1-50

Move 8-89, 24-48, 24-48, 24-48, 24-60, 24-60, 24-60, 24-65, 24-65, 24-65, 26-114, 26-114, 26-114, 26-155, 26-204

MoveCSToEnd 8-160

MoveEnityToGroup 8-160

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

MoveFaces 8-161

MovePlotCurvestoGroup 10-95

MovePlotCurvestoNewGroup 10-95

MultipleEdit 23-3

N

Named Arguments 1-57

NdExplorer

Script Commands 19-1

NetClass Methods 24-140

Network Data Explorer Manager Commands 25-176

NewProject 4-41

Normalize 24-61

O

oAnsoftApp object 1-50

Object Identifiers and Script Recording 24-2

Object Movement and Modification Methods 24-42

object-oriented scripting 2-1

oDesign object 1-50

oDesktop object 1-50

oEditor object 1-50

OffsetFaces 8-90

oModule object 1-50

Open 19-40

OpenAndConvertProject 4-41

OpenProject 4-42, 26-205

OpenWindowForAllReports 10-96

OpenWindowForReports 10-97

oProject object 1-50

Optimetrics module commands

DeleteSetups 11-8, 11-31

DistributeAnalyzeSetup 11-8, 11-32

EnableSetup 11-9, 11-32

ExportDXConfigFile 11-10, 11-33

ExportOptimetricsProfile 11-10, 11-34

ExportOptimetricsResults 11-11, 11-34

ExportParametricResults 11-12, 11-35

ExportRespSurfaceMinMaxTable 11-13, 11-36

ExportRespSurfaceRefinePoints 11-14, 11-37

ExportRespSurfaceResponsePoints 11-15, 11-38

ExportRespSurfaceVerificationPoints 11-15, 11-38

GetChildNames 11-16, 11-17, 11-43, 11-44

GetOptimetricResult 11-20, 11-46

GetOptimetricsResults 11-20

GetSetupNames 11-22, 11-48

GetSetupNamesByType 11-23, 11-49

ImportSetup 11-24, 11-49

RenameSetup 11-25, 11-51

SolveSetup 11-7, 11-26, 11-27, 11-30, 11-52, 11-53

Optimetrics Module Script Commands 11-1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

optimization commands
EditSetup 11-61
InsertSetup 11-65
Optimization Script Commands 11-61
Other Creation Methods 24-24
Other oEditor Commands 8-183
output variable commands
CreateOutputVariable 9-1
DeleteOutputVariable 7-35, 9-3
DoesOutputVariableExist 9-3
EditOutputVariable 7-43, 9-4
GetOutputVariableValue 7-81, 9-7
Output variable commands
ExportOutputVariables 9-5
ImportOutputVariables 9-8
Output Variable Script Commands 9-1
OverlayCurrents 7-91
OverlayFarField 7-92
OverlayMesh 7-92
OverlayNearField 7-93

P

Padstack Manager Script Commands 25-183
PageSetup 8-233
PageSetup (Layout Editor) 4-43, 24-222, 26-114
Pan 26-156
parametric commands
EditSetup 11-55
GenerateVariationData Parametric 11-16, 11-42, 11-56
InsertSetup 11-28, 11-54, 11-56, 11-60
Parametric Script Commands 11-54
Paste 6-49, 8-255, 8-256, 24-48, 26-156
Paste (Model Editor) 8-91
Paste (Project Object) 6-49
PasteDesign 7-93
PasteItemCommand 7-94
PastePlotSettings 10-97
PasteRadFieldSetup 15-8
PasteReports 10-98
PasteReportsWithLegacyNames 10-98
PasteSetup
Optimetrics module command 11-24, 11-50
PasteTraces 10-99
PasteTracesWithLegacyNames 10-99
PauseRecording 4-44
PauseScript 4-44
Point (Layout Editor) 24-57, 26-115
Point Object 24-57
PointInPolygon 24-71
Polygon 24-57, 26-115

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Polygon Object 24-62

Port Methods 24-156

PositionRelative 24-84, 26-115

Print 4-45

Project object commands

AddDataset 6-4, 7-7, 25-6

AddMaterial 6-6, 25-10, 25-41

AreMaterialPropertiesEqual 25-28

AreSurfaceMaterialPropertiesEqual 25-29

ChangeProperty 25-30, 26-134

CloneMaterial 6-14, 25-40

Close 6-15

CopyDesign 6-15

CutDesign 6-16

DeleteDataset 6-17, 7-32, 25-40

DeleteDesign 6-17

EditDataset 6-18, 6-27, 6-44, 7-37, 7-52, 7-88, 25-54, 25-68, 25-74

EditMaterial 6-20, 25-56

ExportMaterial 6-28, 25-69

GetActiveDesign 6-28

GetArrayVariables 2-77, 6-29

GetChildObject Design 6-30, 7-68

GetChildObject Modeler 8-202

GetChildObject Radiation 15-5

GetDependentFiles 6-32

GetDesign 2-59

GetDesignID 7-69

GetDesignName 7-70

GetDesignType 7-71

GetName 6-35

GetObjPath 6-36

GetPath 6-36

GetProjectMaterialNames 25-72

GetProperties 2-78, 6-40, 7-83, 24-218, 26-106, 26-153, 26-204

GetPropertyValue 2-79, 6-41, 7-84, 8-223, 10-75, 24-219, 25-73, 26-106, 26-153

GetPropEvaluatedValue 6-37, 8-223

GetPropNames Modeler 8-225

GetPropSIValue 6-39, 8-226

GetPropvalue Modeler 8-227

GetPropvalue Optimetrics 11-22, 11-47

GetPropvalue Project 6-40, 7-86

GetPropvalue Radiation 15-6

GetPropvalue Reporter 10-77

GetSolutionVariation 19-36

GetTopDesignList 6-43

GetTopEntryValue 13-47, 14-24

GetVariables 2-80, 6-44, 7-87

GetVariableValue 2-81, 6-43, 7-88

InsertDesign 6-46, 7-90

LoadSolution 19-39

Paste 6-49, 6-49

Redo 6-50

RemoveAllUnusedDefinitions 6-50

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveMaterial 6-51, 25-78
RemoveUnusedDefinitions 6-51, 25-81
Rename 6-52
RestoreProjectArchive 4-47
Save 6-54
SaveAs 6-54
SaveAsStandAloneProject 6-56
SaveProjectArchive 6-57
SetActiveDesign 6-58
SetPropertyValue 2-81, 6-60, 7-113, 24-226, 25-82, 26-208
SetPropValue Design 7-112, 15-9
SetPropValue Optimetrics 11-26, 11-51
SetPropValue Project 6-59, 10-103
SetVariableValue 2-83, 6-61, 7-115
SimulateAll 6-9, 6-62
Undo 6-62
UpdateDefinitions 6-63, 25-88
Project Object Script Commands 6-1
ProjectSheet 8-163
property commands
ChangeProperty 25-30, 26-134
GetArrayVariables 2-77, 6-29
GetProjectMaterialNames 25-72
GetProperties 2-78, 6-40, 7-83, 24-218, 26-106, 26-153, 26-204
GetPropertyValue 2-79, 6-41, 7-84, 8-223, 10-75, 24-219, 25-73, 26-106, 26-153
GetPropEvaluatedValue 6-37, 8-223
GetPropSIValue 6-39, 8-226
GetTopEntryValue 13-47, 14-24
GetVariables 2-80, 6-44, 7-87
GetVariableValue 2-81, 6-43, 7-88
SetPropertyValue 2-81, 6-60, 7-113, 24-226, 25-82, 26-208
SetVariableValue 2-83, 6-61, 7-115
Property Script Commands 2-25
Conventions uSed in thie Chapter 2-36
Object Script Property Function Summary 2-27
PropertyExists 2-67
PropHost Functions 2-48
PullExcitationsFromTransient 21-41
PurgeHistory 8-164
PushExcitations 7-94, 24-223, 26-118
Layout Editor 7-94, 24-223, 26-118

Q

QuitApplication 4-45

R

Radiation boundary
Infinite Sphere 16-1, 16-4

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Near Field Line 16-7, 16-9
Near Field Plane 16-10, 16-13
Near Field Sphere 16-16, 16-19
Radiation module commands
AddAntennaOverlay 15-10
AddRadFieldSourceGroup 15-12
CopyRadFieldSetup 15-2
DeleteFarFieldSetup 15-2
DeleteNearFieldSetup 15-3
DeleteSetup 15-4
EditAntennaArraySetup 15-45
EditAntennaOverlay 15-13
EditBoxSetup 15-14
EditFarFieldSphereSetup 15-16
EditInfiniteSphereSetup 15-17
EditLineSetup 15-19
EditNearFieldBoxSetup 15-20
EditNearFieldLineSetup 15-22
EditNearFieldRectangleSetup 15-23
EditNearFieldSphereSetup 15-24
EditRadFieldSourceGroup 15-26
EditRadiatedPowerCalculationMethod 15-4
EditRectangleSetup 15-27
EditSphereSetup 15-28
ExportElementPatternToFile 15-50
ExportFieldsToFile 15-51
ExportParametersToFile 15-53
ExportRadiationFieldsToFile 15-54
ExportRadiationParametersToFile 15-55
GetSetupNames 15-7
InsertBoxSetup 15-30
InsertFarFieldSphereSetup 15-32
InsertInfiniteSphereSetup 15-33
InsertLineSetup 15-35
InsertNearFieldBoxSetup 15-36
InsertNearFieldLineSetup 15-38
InsertNearFieldRectangleSetup 15-39
InsertRectangleSetup 15-42
InsertSphereSetup 15-40, 15-43
PasteRadFieldSetup 15-8
RenameSetup 15-8
Radiation Module Script Commands 15-1, 16-1
radiation script commands 15-2
ReassignFieldPlot 13-52
Redo 7-97, 26-157
design-level command 7-96
project-level command 6-50
RefreshJobMonitor 4-46
RefreshMeshOverlays 21-44
ReleaseData 10-92
Remove 25-76, 25-77, 25-79, 25-80, 25-128, 25-155, 25-175, 25-203, 25-229

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveAllUnusedDefinitions 6-50

RemoveBadEdges 8-235

RemoveBadFaces 8-235

RemoveBadVertices 8-236

RemoveCables 8-259

RemoveImportData 7-97

RemoveLayer 26-120, 26-205

RemoveLevels 26-157

RemoveMaterial 6-51, 25-78

RemoveModelingProperties 7-97

RemovePort 24-176, 26-158, 26-205

RemovePortsFromAllNets 24-177, 26-121

RemovePortsFromNet 24-177, 26-122

RemovePortsOnComponents 24-178, 26-122

RemoveProp 2-67

RemoveSolverOnDemandModel 25-128

RemoveUnusedDefinitions 6-51, 25-81

Rename 6-52, 19-41, 21-44, 22-27, 23-3

Rename [Interface Port] 22-27

RenameDesignInstance 7-98, 7-98

RenameFieldPlot 13-53

RenameImportData 7-99

RenamePart 8-236

RenamePlotFolder 13-54

RenameReport 10-100

RenameSetup

Optimetrics module command 11-25, 11-51

Radiation module command 15-8

RenameSource [Interface Source] 7-99, 22-27

RenameTraces 10-101

Renormalize 19-42

Reorder 19-43, 19-44

Replacewith3DComponent 8-164

Report module commands

GetChildNames 10-68

Reporter editor commands

AddAllEyeMeasurements 10-6

AddCartesianLimitLine 10-6

AddCartesianLimitLineFromCurve 10-7

AddCartesianLimitLineFromEquation 10-9

AddCartesianXMarker 10-10, 10-11

AddCartesianYMarkerToStack 10-11

AddDeltaMarker 10-12

AddMarker 10-13

AddMarkerToPlot 13-4

AddNote 10-14

AddTraceCharacteristics 10-15

AddTraces 10-17

ApplyReportTemplate 10-18

ChangeProperty 7-12, 10-19

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ClearAllMarkers 10-26

ClearAllTraceCharacteristics 10-26

CloneReportsFromDatasetSolution 10-27

CopyPlotSettings 10-28

CopyReportDefinition 10-28

CopyReportsData 10-29

CopyTraceDefinitions 10-29

CopyTracesData 10-30

CreateReport 7-19, 10-31

CreateReportFromTemplate 10-43

CreateReportOfAllQuantities 10-44

DeleteAllReports 10-45

DeleteReport 10-46

DeleteTraces 10-46

DoesSupportTraceCharacteristics 10-47

EditCartesianXMarker 10-48

EditCartesianYMarker 10-49

EditMarker 10-49

ExportEyeMaskViolation 10-50

ExportImageToFile 10-51

ExportPlot3DToFile 10-59

ExportPlotImageToFile 13-41, 13-42

ExportReport 7-64, 10-60

ExportReportDataToFile 10-61

ExportTableToFile 10-61

ExportToFile 10-62, 10-93

ExportUniformPointsToFile 10-63

FFTOnReport 12-15

GetAllCategories 10-64

GetAllQuantities 10-65

GetAllReportNames 10-66

GetAvailableDisplayTypes 10-67

GetAvailableReportTypes 10-67

GetAvailableSolutionss 10-68

GetChildTypes ReportSetup 10-69

GetCurvePropServerName 10-70

GetDisplayType 10-71

GetDynLinkIntrinsicVariables 10-71

GetDynLinkQtyValueState 10-72

GetDynLinkTraces 10-72

GetDynLinkVariableValues 10-73

GetPropNames 10-76

GetQtyExpressionsForSourceTrace 10-77

GetReportSummaryForRegressionTesting 10-79

GetReportTraceNames 10-78

GetSolutionContexts 10-79, 10-80

GroupPlotCurvesByGroupingStrategy 10-93

ImportReportDataIntoReport 10-94

MovePlotCurvestoGroup 10-95

MovePlotCurvestoNewGroup 10-95

OpenWindowForAllReports 10-96

OpenWindowForReports 10-97

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PastePlotSettings 10-97

PasteReports 10-98

PasteReportsWithLegacyNames 10-98

PasteTraces 10-99

PasteTracesWithLegacyNames 10-99

RenameReport 10-100

RenameTraces 10-101

ResetPlotSettings 10-101

SavePlotSettingsAsDefault 10-102

SetLinkOutputTraces 10-102

UnGroupPlotCurvesInGroup 10-104

UpdateAllReports 10-105

UpdateReports 10-105

UpdateTraces 10-106

UpdateTracesContextAndSweeps 10-108

Reporter Editor Script Commands 10-1

Reporter module commands

GetChildObject 10-69

ReportTemplates 7-100

Reset 19-45

ResetLogging 4-47

ResetPlotSettings 10-101

RestoreWindow 4-48

ResumeRecording 4-49

Rotate 8-92, 24-48, 24-61, 24-65, 26-123, 26-158, 26-206

Running Instance Manager Script Commands 5-1

RunProgram 4-50

RunScript 4-50

RunScriptWithArguments 4-52

RunToolkit 6-53, 7-100

S

sample scripts

variable helix 28-6

Save 6-54, 26-159, 26-206

SaveAs 6-54

SaveAsStandAloneProject 6-56

SaveFieldsPlots 13-55

SaveNamedExpressions 13-55, 14-25

SavePlotSettingsAsDefault 10-102

Scale 8-93, 24-65

Script and Library Scripts 25-204

script commands

field overlay 13-1

radiation 15-2

Script Commands for Creating and Modifying Radiation Setups 15-9

Script Commands for Exporting Antenna Parameters and Max Field Parameters 15-50

Script Commands for Modifying Antenna Array Setups 15-45

scripting

object-oriented 2-1

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

scripts
CPython 1-44
IronPython 1-2
overview 1-1
pausing 1-54
recording 1-54
running 1-53
stopping 1-53
Scripts and Locked Layers 1-60
Section 8-166
SelectAll 24-225, 26-123, 26-159
SelectInLayout 22-28, 23-3
SelectScheduler 4-14, 4-53
SendToBack 26-160
sensitivity commands
EditSetup 11-70
Sensitivity Script Commands 11-70
SeparateBody 8-168
Set 24-58
SetActiveDefinitionEditor 6-58, 26-207
SetActiveDesign 6-58
SetActiveEditor 7-101, 7-101, 7-103
SetActiveLevel 26-160
SetActiveProject 4-54
SetActiveProjectByPath 4-55
SetAllPortImpedance 19-46
SetArc (Layout Editor) 24-59
SetCallback 2-68
SetClosed 24-64
SetCS 24-87, 26-124
SetDescription 2-69
SetFastTrans-formationForLayoutComponent 7-109, 7-110
SetFieldPlotSettings 13-56
SetHidden 2-69
SetInitialLevels 26-161
SetLayerMapping 24-138, 26-124
SetLibraryDirectory 4-57
SetLinkOutputTraces 10-102
SetModelUnits 8-168
SetPlotFolderSettings 13-58
SetPortDeembedDistance 19-47
SetPortImpedance 19-47
SetPostProcSettings 19-48
SetProjectDirectory 4-58
SetPropertyValue 2-81, 6-60, 7-113, 24-226, 25-82, 26-162, 26-208
SetPropValue Design 7-112, 15-9
SetPropValue Modeler 8-237
SetPropValue Optimetrics 11-26, 11-51
SetPropValue Project 6-59, 10-103
SetReadOnly 2-70
SetShowLayoutForLayoutComponent 7-114
SetTempDirectory 4-61
SetText 2-70
Setting Numerical Values 1-59

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetTopDownViewDirectionForActiveView 8-242

SetTopDownViewDirectionForAllViews 8-242

SetValue 2-71

SetVariableValue 2-83, 6-61, 7-115

SetVisibleLevels 26-162

SetWCS 8-169

SetX 24-58

SetY 24-59

SGetAppDesktop 3-1

ShowAntennaParameterOverlay 13-62

ShowIWindow 8-170

Simplify 8-171

SimulateAll 6-9, 6-62

Simulation Setup Commands 21-1

SimValueContext 9-9

Sleep 4-62

Smooth 19-49

Solution module commands

- GetSourceContexts 12-19

Solutions module commands

- DeleteFieldVariation 7-33
- DeleteFullVariation 7-34
- DeleteImportData 12-1
- DeleteSolutionVariation 7-35, 12-2
- EditQ3DSources 12-3
- EditSources 12-6
- ExportEigenmodes 12-10
- ExportForHSpice 7-53, 12-11
- ExportNMF 12-14
- GetAdaptiveFreq 12-16
- GetAvailableVariations 12-17
- GetSolutionVersionID 12-18
- GetSolveRangeInfo 12-18
- GetValidISolutionLIst 12-20
- HasFields 12-20
- HasMatrixData 12-21
- ImportSolution 12-22
- ImportTable 12-23
- IsFieldAvailableAt 12-24
- ListMatchingVariations 12-25
- ListValuesOfVariable 12-25
- ModuleHasMesh 12-21

Solutions Module commands

- ExportNetworkData 12-12

Solutions Module Script Commands 12-1

SolveAllSetup, Optimetrics module command 11-26, 11-52

SolveSetup, Optimetrics module command 11-27, 11-53

Split 8-173

SplitRegion 24-77

statistical commands

- EditSetup 11-74
- InsertSetup 11-75

Statistical Script Commands 11-74

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Stitch 8-175

StitchLines 24-227, 26-126

StopSimulations 4-63

Stretch 19-51

Structures 1-30

SubmitJob 4-63

Subtract 8-176, 24-73, 24-73, 24-77, 24-77, 26-127, 26-127, 26-208

SweepAlongPath 8-71

SweepAlongVector 8-72

SweepAroundAxis 8-74

SweepFacesAlongNormal 8-75, 8-75, 8-177, 8-177

SweepFacesAlongNormalWithAttributes 8-77

Symbol Editor Scripts 26-130

Symbol Manager Script Commands 25-209, 25-230

T

Terminate 19-52

ThickenSheet 8-178

ToggleLevel 26-163

ToggleViaPin 24-78, 26-127, 26-163, 26-209

U

UDM Structures 1-30

UncoverFaces 8-179

Undo 7-116, 26-164

project-level command 6-62

Ungroup 8-182, 24-87, 26-128

UnGroupPlotCurvesInGroup 10-104

Unite 8-180, 24-72, 24-72, 24-76, 24-76, 26-128, 26-128, 26-209

UnselectAll 24-228, 26-129

UpdateAllFieldsPlots 13-62

UpdateAllReports 10-105

UpdateCableHarness 8-259

UpdateComponentDefinition 8-78

UpdateDefFromBlock 25-83

UpdateDefFromBlockEx 25-85

UpdateDefinitions, project-level command 6-63, 25-88

UpdateDynamicLink 25-130

UpdatePriorityList 8-243

UpdateQuantityFieldsPlots 13-63

UpdateReports 10-105

UpdateTraces 10-106

UpdateTracesContextAndSweeps 10-108

UpgradeVersion 8-244

User defined documents module commands

GetDocumentNames 17-5

User defined solution module commands

CreateUserDefinedSolution 18-1

DeleteUserDefinedSolutions 18-2

EditUserDefinedSolution 18-3

GetUserDefinedSolutionNames 18-4

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetUserDefinedSolutionProperties 18-4

User Defined Solutions Commands 18-1

V

Validate3DComponent 8-245

ValidateCircuit 7-116

variables

hierarchy in HFSS 1-50

used in HFSS scripts 1-50

W

WrapSheet 8-182

WriteHistoryTreeLayoutForTest 8-245

X

Xor 24-73

Z

ZoomArea 26-164

ZoomIn 26-165

ZoomOut 26-166

ZoomPrevious 26-166

ZoomToFit 24-228, 26-129, 26-167

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.