

2026R1

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

Icepak® 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 1-6
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 1-9
Create a Cone using IronPython 1-10
Creating User Defined Primitives and User Defined Models in Python Scripts 1-14
Advantages Compared to C++ 1-14
Changes compared to C 1-14
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 1-15
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 1-19
Edit/Reload 1-20
UDPExtension 1-20
IUDPExtension Methods 1-20
Mandatory Methods 1-20
GetLengthParameterUnits() 1-20
Optional Methods 1-21
Example UDP 1-22
UDMExtension 1-22
Import 1-22
Main class: UDMExtension 1-22
IUDMExtension Methods 1-23
Mandatory Methods 1-23
Optional Methods 1-24
Example UDM 1-25
UDMFunctionLibrary 1-26
Functions list: 1-26
UDM/UDP Functions 1-27
Return Values for Each UDM and UDP Function 1-27
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 ... 1-37
Enum constants: ... 1-38
UDP Python Example ... 1-40
Introduction to CPython ... 1-44
Creating an External Script ... 1-44
Start ansysedt as GRPC server ... 1-45
Connect Functions: ... 1-45
Example: ... 1-46
Launching Electronics Desktop ... 1-46
Connecting with a Running Instance of Electronics Desktop ... 1-46
Closing Electronics Desktop/Ending the Script ... 1-47
-grpcsrv Flag ... 1-47
Ansys Electronics Desktop Scripting ... 1-49
Overview of Electronics Desktop Scripting Objects ... 1-49
oAnsoftApp ... 1-50
oDesktop ... 1-50
oProject ... 1-50
oDesign ... 1-50
oEditor ... 1-50
oModule ... 1-51
Example Script Opening ... 1-51
GetActiveProject and GetActiveDesign for Wider Use ... 1-52
Running a Script ... 1-52
Within Electronics Desktop ... 1-52
From the Command Line ... 1-53
Recording a Script ... 1-53
Recording a Script to File ... 1-53
Recording a Script to a Project ... 1-54
Working with Project Scripts ... 1-55
Executing a Script from Within a Script ... 1-56

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 ... 1-56
Named Arguments ... 1-56
IronPython Example ... 1-57
Setting Numerical Values ... 1-58
Layout Scripts and the Active Layer ... 1-58
Scripts and Locked Layers ... 1-59
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
Additional Details Specific to Icepak Monitor and Mesh Modules ... 2-16
3D Component Encapsulation ... 2-17
Additional Details on Solve Setup ... 2-17
Object-Oriented Scripting for Editors ... 2-18
Schematic Editor ... 2-18
Object-Oriented Scripting for Materials ... 2-19
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 2-29
3D Modeler Object 2-30
Variable Object 2-31
Optimetrics Module Object: 2-32
Optimetrics Setup Object 2-32
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
GetArrayVariables 2-38
GetProperties 2-39
GetPropertyValue 2-40
GetVariables 2-41
GetVariableValue 2-42
SetPropertyValue 2-43
SetVariableValue 2-44
3 - Application Object Script Commands 3-1
GetAppDesktop 3-2
4 - Desktop Object Script Commands 4-1
AddMessage 4-6
ClearMessages 4-7
CloseAllWindows 4-10
CloseProject 4-11
CloseProjectNoForce 4-12
Count 4-12
DeleteProject 4-13
DownloadJobResults 4-14
DeleteRegistryEntry 4-15
DoesRegistryValueExist 4-15

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

EnableAutoSave ... 4-16
ExportOptionsFiles ... 4-17
GetActiveProject ... 4-17
GetAutoSaveEnabled ... 4-18
GetBuildDateTimeString ... 4-18
GetCustomMenuSet ... 4-19
GetDefaultUnit ... 4-20
GetDesktopConfiguration ... 4-22
GetDistributedAnalysisMachines ... 4-22
GetDistributedAnalysisMachinesForDesignType ... 4-23
GetExeDir ... 4-23
GetGDIObjectCount ... 4-24
GetLibraryDirectory ... 4-24
GetLocalizationHelper ... 4-25
GetMessages ... 4-26
GetPersonalLibDirectory ... 4-27
GetProcessID ... 4-27
GetProjectDirectory ... 4-28
GetProjectList ... 4-28
GetProjects ... 4-29
GetRegistryInt ... 4-29
GetRegistryString ... 4-30
GetRunningInstancesMgr ... 4-31
GetSchematicEnvironment ... 4-31
GetScriptingToolsHelper ... 4-32
GetSysLibDirectory ... 4-33
GetTempDirectory ... 4-33
GetUserLibDirectory ... 4-34
GetVersion ... 4-34
IsFeatureEnabled ... 4-35

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

KeepDesktopResponsive 4-36
LaunchJobMonitor 4-37
NewProject 4-37
OpenProject 4-38
OpenProjectWithConversion 4-38
PauseRecording 4-39
PauseScript 4-39
Print 4-40
QuitApplication 4-40
RefreshJobMonitor 4-41
ResetLogging 4-42
RestoreProjectArchive 4-42
RestoreWindow 4-43
ResumeRecording 4-44
RunACTWizardScript 4-44
RunProgram 4-45
RunScript 4-45
RunScriptWithArguments 4-47
SelectScheduler 4-48
SetActiveProject 4-49
SetActiveProjectByPath 4-50
SetCustomMenuSet 4-50
SetDesktopConfiguration 4-51
SetLibraryDirectory 4-52
SetProjectDirectory 4-53
SetRegistryFromFile 4-53
SetRegistryInt 4-54
SetRegistryString 4-55
SetSchematicEnvironment 4-55
SetTempDirectory 4-56

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

ShowDockingWindow 4-57
Sleep 4-57
SubmitJob 4-58
TileWindows 4-59
Desktop Commands For Registry Values 4-60
DeleteRegistryEntry 4-61
DoesRegistryValueExist 4-61
GetRegistryInt 4-62
GetRegistryString 4-63
SetRegistryFromFile 4-63
SetRegistryInt 4-64
SetRegistryString 4-65
ImportExport Tool Commands 4-65
ImportANF 4-66
ImportANFv2 4-67
ImportAutoCAD 4-68
ImportAWRMicrowaveOffice 4-68
ImportEDB 4-69
ImportExtracta 4-70
ImportGDSII 4-71
ImportGerber 4-72
ImportIDF 4-72
ImportIDFandMerge 4-75
ImportIDX 4-76
ImportIPC 4-78
ImportODB 4-79
ImportXFL 4-80
ImportECXML 4-80
ImportJEDEC 4-81
Component 2.0 (Beta) Commands 4-81

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

CreateComponent2_0 4-82
EditComponent2_0 4-90
InsertComponent2_0 4-98
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
ClearMessages 6-10
Close 6-10
CopyDesign 6-11
CutDesign 6-12
DeleteDataset 6-12
DeleteDesign 6-13
DeleteToolObject 6-14
EditDataset 6-14
EditMaterial 6-16
ExportDataset 6-23
ExportMaterial 6-24
GetActiveDesign 6-24
GetArrayVariables 6-25
GetChildNames [Project] 6-26
GetChildObject [Project] 6-26
GetChildTypes [Project] 6-27
GetDefinitionManager 6-28
GetDependentFiles 6-28
GetDesign 6-29

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

GetDesigns ... 6-29
GetEDBHandle ... 6-30
GetLegacyName ... 6-31
GetName [Project] ... 6-31
GetObjPath [Project] ... 6-32
GetPath ... 6-32
GetPropNames [Project] ... 6-33
GetPropValue [Project] ... 6-34
GetProperties ... 6-34
GetPropertyValue ... 6-35
GetTopDesignList ... 6-37
GetVariableValue ... 6-37
GetVariables ... 6-38
HasDataset ... 6-39
ImportDataset ... 6-39
Note About File Types: ... 6-40
InsertDesign ... 6-41
InsertDesignWithWorkflow ... 6-42
InsertToolObject ... 6-43
Paste (Project Object) ... 6-43
Redo [Project Level] ... 6-44
RemoveAllUnusedDefinitions ... 6-44
RemoveMaterial ... 6-45
RemoveUnusedDefinitions ... 6-46
Rename ... 6-47
Save ... 6-47
SaveAs ... 6-48
SaveAsStandAloneProject ... 6-50
SaveProjectArchive ... 6-51
SetActiveDefinitionEditor ... 6-52

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

SetActiveDesign ... 6-52
SetPropValue [Project] ... 6-53
SetPropertyValue ... 6-54
SetVariableValue ... 6-55
SimulateAll ... 6-56
Undo [Project] ... 6-56
UpdateDefinitions ... 6-57
7 - Design Object Script Commands ... 7-1
AddDataset ... 7-4
AddModelingProperties ... 7-6
AnalyzeAll [design] ... 7-6
AnalyzeAllNominal ... 7-7
ApplyMeshOps ... 7-8
ConstructVariationString ... 7-8
CopyItemCommand ... 7-9
CreatePcbPowermap ... 7-9
CreateReport ... 7-10
DeleteDataset ... 7-23
DeleteOutputVariable ... 7-24
DeleteSolutionVariation ... 7-24
DeleteOutputVariable ... 7-25
EditCoSimulationOptions ... 7-26
EditDesignSettings ... 7-26
EditInfiniteArray ... 7-27
EMDesignOptions ... 7-28
ExportConvergence ... 7-29
ExportDataset ... 7-30
ExportLPVROM ... 7-30
EcxmlExport ... 7-31
ExportNetworkCSV ... 7-32

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

ExportProfile 7-32
FitSelected 7-33
GenerateConductivityFile 7-33
Generate Mesh 7-34
GetChildNames [Design] 7-34
GetChildObject [Design] 7-35
GetChildTypes [Design] 7-36
GetDesignID 7-36
GetDesignType 7-37
GetEdgePositionAtNormalizedParameter 7-37
GetGeometryIdsForNetLayerCombinations 7-38
GetGeometryIdsForAllNetLayerCombinations 7-39
GetManagedFilesPath 7-40
GetModule 7-41
GetName 7-42
GetNominalVariation 7-42
GetNoteText 7-43
GetObjPath [Design] 7-43
GetOutputVariableValue 7-44
GetOutputVariables 7-45
GetPostProcessingVariables 7-45
GetProperties 7-46
GetPropertyValue 7-47
GetPropNames [Design] 7-48
GetPropValue [Design] 7-49
GetSelections [Design] 7-50
GetSolutionType 7-50
GetSolveInsideThreshold 7-50
GetVariables 7-51
GetVariableValue 7-51

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

GetVariationVariableValue 7-52
HideGeometry 7-53
ImportDataset 7-53
Note About File Types: 7-54
ImportNetworkCSV 7-55
InitializeSystemCoupling 7-56
InsertDesign 7-57
PasteDesign (Design Object) 7-57
Redo [Design] 7-58
RenameDesignInstance 7-59
RenameSource [Interface Source] 7-59
RevertAllToInitialCondition 7-60
RunCTMAPI 7-60
Settings.toml File 7-63
General Settings 7-64
Operation Settings 7-65
Elitism Settings 7-65
RunToolkit 7-65
SARSetup 7-66
SelectObjects 7-67
SetActiveEditor 7-67
SetAllowMaterialOverride 7-68
SetBackgroundMaterial 7-69
SetLengthSettings 7-69
SetPropValue [Design] 7-70
SetPropertyValue 7-71
SetSolutionType 7-72
SetSolveInsideThreshold 7-74
SetSourceContexts 7-74
SetVariableValue 7-75

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

ShowGeometry ... 7-76
Undo [Design] ... 7-76
ValidateDesign ... 7-77
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
CreateBondwire ... 8-6
CreateBox ... 8-9
CreateCircle ... 8-11
CreateCone ... 8-13
CreateCutplane ... 8-15
CreateCylinder ... 8-17
CreateEllipse ... 8-19
CreateEquationCurve ... 8-21
CreateEquationSurface ... 8-24
CreateHelix ... 8-26
CreatePoint ... 8-28
CreatePolyline ... 8-29
CreateRectangle ... 8-32
CreateRegularPolygon ... 8-34
CreateRegularPolyhedron ... 8-36
CreateSpiral ... 8-38
CreateTorus ... 8-40
CreateUserDefinedModel ... 8-42
CreateUserDefinedPart ... 8-49
Edit3DComponent ... 8-54
Edit3DComponentDefinition ... 8-56
EditNativeComponentDefinition [Beta – Layout Components in Icepak] ... 8-57

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

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
InsertNativeComponent [Beta – Layout Component in Icepak] ... 8-69
InsertPolylineSegment ... 8-73
SweepAlongPath ... 8-75
SweepAlongVector ... 8-76
SweepAroundAxis ... 8-78
SweepFacesAlongNormal ... 8-80
SweepFacesAlongNormalWithAttributes ... 8-81
UpdateComponentDefinition ... 8-83
Edit Menu Commands ... 8-83
Copy ... 8-84
DeletePolylinePoint ... 8-85
DuplicateAlongLine ... 8-86
DuplicateAroundAxis ... 8-88
DuplicateMirror ... 8-89
Mirror ... 8-92
Move ... 8-93
OffsetFaces ... 8-94
Paste (Model Editor) ... 8-95
Rotate ... 8-96
Scale ... 8-97
Modeler Menu Commands ... 8-98

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

AlignFaces ... 8-100
AssignMaterial ... 8-101
Chamfer ... 8-103
CleanUpModel ... 8-104
Connect ... 8-104
CoverLines ... 8-105
CoverSurfaces ... 8-106
CreateEntityList ... 8-106
CreateFaceCS ... 8-108
CreateGroup ... 8-111
CreateNamedSelection ... 8-113
CreateObjectCS ... 8-114
CreateObjectFromEdges ... 8-118
CreateObjectFromFace ... 8-120
CreateObjectFromFaces ... 8-121
CreateRelativeCS ... 8-122
DeleteEmptyGroups ... 8-124
DeleteLastOperation ... 8-125
DeleteOperation ... 8-125
DetachEdges ... 8-127
DetachFaces ... 8-128
EditEntityList ... 8-129
EditFaceCS ... 8-130
EditNamedSelection ... 8-134
EditObjectCS ... 8-135
EditRelativeCS ... 8-140
Export ... 8-141
ExportModelImageToFile ... 8-143
ExportModelMeshToFile ... 8-146
Fillet ... 8-147

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

FlattenGroup 8-148
GenerateHistory 8-149
HealObject 8-150
Import 8-152
ImportDXF [Modeler] 8-155
ImportFromClipboard 8-157
ImportGDSII [Modeler] 8-158
Imprint 8-160
ImprintProjection 8-161
Intersect 8-163
MoveCStoEnd 8-164
MoveEntityToGroup 8-164
MoveFaces 8-165
ProjectSheet 8-167
PurgeHistory 8-168
ReplaceWith3DComponent 8-168
Section 8-170
SeparateBody 8-172
SetModelUnits 8-172
SetWCS 8-173
ShowWindow 8-174
Simplify 8-175
Split 8-177
Stitch 8-179
Subtract 8-180
SweepFacesAlongNormal 8-181
ThickenSheet 8-182
UncoverFaces 8-183
Unite 8-184
Ungroup 8-186

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

WrapSheet ... 8-186
Other oEditor Commands ... 8-187
AddDefinitionFromBlock ... 8-190
AddDefinitionFromLibFile ... 8-192
AddViewOrientation ... 8-195
AssignSurfaceMaterial ... 8-199
BreakUDMConnection ... 8-201
CloseAllWindows[Editor] ... 8-202
Defeature ... 8-203
Delete ... 8-203
FitAll ... 8-204
GenerateAllUserDefinedModels ... 8-205
GenerateUserDefinedModel ... 8-205
GeometryCheckAndAutofix ... 8-206
GetBodyNamesByPosition ... 8-207
GetChildNames [Modeler] ... 8-208
GetChildObject [Modeler] ... 8-211
GetChildTypes [Modeler] ... 8-212
GetEdgeByPosition ... 8-214
GetEdgeIDFromNameForFirstOperation ... 8-215
GetEdgeIDsFromFace ... 8-216
GetEdgeIDsFromObject ... 8-216
GetEdgeLength ... 8-217
GetEntityIDsContainedByNamedSelection ... 8-217
GetEntityListIDByName ... 8-218
GetExtendedDefinitionObject ... 8-218
GetFaceArea ... 8-219
GetFaceCenter ... 8-219
GetFaceByPosition ... 8-220
GetFaceIDFromNameForFirstOperation ... 8-221

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

GetFaceIDs 8-222
GetFaceIDsOfSheet 8-222
GetGeometryModelerMode 8-223
GetGroupSubmodelNames 8-223
GetMatchedObjectName 8-224
GetModelBoundingBox 8-225
GetModelUnits 8-225
GetNumObjects 8-226
GetObjectIDByName 8-226
GetObjectName 8-227
GetObjectNameByEdgeID 8-227
GetObjectNameByFaceID 8-228
GetObjectNameByID 8-228
GetObjectNameByVertexID 8-229
GetObjectsByMaterial 8-229
GetObjectShapeType 8-230
GetObjectsInGroup 8-231
GetObjectVolume 8-231
GetObjPath [Editor] 8-232
GetPartsForUserDefinedModel 8-233
GetPoints [3D Modeler Editor] 8-233
GetPropEvaluatedValue 8-234
GetPropertyValue 8-234
GetPropNames [Modeler] 8-236
GetPropSIValue 8-237
GetPropValue [Modeler] 8-238
GetRelativeCoordinateSystems 8-239
GetSelections [Model Editor] 8-239
GetSubGroupsInGroup 8-240
GetUserPosition 8-240

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

GetVertexIDFromNameForFirstOperation ... 8-241
GetVertexIDsFromEdge ... 8-242
GetVertexIDsFromFace ... 8-242
GetVertexIDsFromObject ... 8-243
GetVertexPosition ... 8-243
GetWireBodyNames ... 8-244
PageSetup ... 8-244
RemoveBadEdges ... 8-246
RemoveBadFaces ... 8-246
RemoveBadVertices ... 8-247
RenamePart ... 8-247
SetPropValue [Modeler] ... 8-248
Setting User-Defined Attributes ... 8-249
SetTopDownViewDirectionForActiveView ... 8-253
SetTopDownViewDirectionForAllViews ... 8-254
UpdatePriorityList ... 8-254
UpgradeVersion ... 8-255
Validate3DComponent ... 8-256
WriteHistoryTreeLayoutForTest ... 8-257
Cable Modeling Commands ... 8-258
AddCableToBundle ... 8-259
CreateCableBundle ... 8-260
CreateCableHarness ... 8-261
CreateClockSource ... 8-263
CreatePWLSource ... 8-264
CreateStraightWireCable ... 8-266
CreateTwistedPairCable ... 8-267
ExportCableLibrary ... 8-268
ImportCableLibrary ... 8-269
RemoveCable ... 8-269

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

UpdateCableHarness ... 8-270
9 - Output Variable Script Commands ... 9-1
CreateOutputVariable ... 9-1
DeleteOutputVariable ... 9-2
DoesOutputVariableExist ... 9-3
EditOutputVariable ... 9-4
ExportOutputVariables ... 9-5
GetOutputVariables ... 9-5
GetOutputVariableValue ... 9-6
ImportOutputVariables ... 9-7
SimValueContext ... 9-7
10 - Reporter Editor Script Commands ... 10-1
AddAllEyeMeasurements ... 10-4
AddCartesianLimitLine ... 10-4
AddCartesianLimitLineFromCurve ... 10-5
AddCartesianLimitLineFromEquation ... 10-7
AddCartesianXMarker ... 10-8
AddCartesianYMarker ... 10-9
AddCartesianYMarkerToStack ... 10-9
AddDeltaMarker ... 10-10
AddMarker ... 10-11
AddNote ... 10-12
AddTraceCharacteristics ... 10-13
AddTraces ... 10-14
ApplyReportTemplate ... 10-16
ClearAllMarkers ... 10-17
ClearAllTraceCharacteristics ... 10-17
CloneReportsFromDatasetSolution ... 10-18
CopyPlotSettings ... 10-19
CopyReportDefinitions ... 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.

CopyReportsData 10-20
CopyTraceDefinitions 10-20
CopyTracesData 10-21
CreateReport 10-22
CreateReportFromTemplate 10-34
CreateReportOfAllQuantities 10-35
DeleteMarker 10-36
DeleteAllReports 10-37
DeleteReports 10-37
DeleteTraceCharacteristics 10-38
DeleteTraces 10-38
DoesSupportTraceCharacteristics 10-39
DumpAllReportsData 10-40
EditCartesianXMarker 10-40
EditCartesianYMarker 10-41
EditMarker 10-42
ExportEyeMaskViolation 10-42
ExportImageToFile [Reporter] 10-43
ExportModelImageToFile 10-47
ExportModelMeshToFile 10-51
ExportPlot3DToFile [Reporter] 10-51
ExportReport 10-52
ExportReportDataToFile 10-53
ExportTableToFile 10-54
ExportToFile 10-55
ExportToFile [Reporter] 10-56
ExportUniformPointsToFile 10-57
GetAllCategories 10-58
GetAllQuantities 10-59
GetAllReportNames 10-60

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

GetAvailableDisplayTypes 10-60
GetAvailableReportTypes 10-61
GetAvailableSolutions 10-61
GetChildNames [Report Setup] 10-62
GetChildObject [Report Setup] 10-62
GetChildTypes [ReportSetup] 10-63
GetCurvePropServerName 10-64
GetDisplayType 10-64
GetDynLinkIntrinsicVariables 10-65
GetDynLinkQtyValueState 10-65
GetDynLinkTraces 10-66
GetDynLinkVariableValues 10-67
GetName 10-67
GetObjPath [Design] 10-68
GetPropertyValue 10-68
GetPropNames [Reporter] 10-70
GetPropValue [Report Setup] 10-70
GetQtyExpressionsForSourceTrace 10-71
GetReportTraceNames 10-71
GetReportSummaryForRegressionTesting 10-72
GetSolutionContexts 10-72
GetSolutionDataPerVariation 10-73
GetDataUnits 10-75
GetDesignVariableNames 10-76
GetDesignVariableUnits 10-76
GetDesignVariableValue 10-77
GetDesignVariationKey 10-78
GetImagDataValues 10-79
GetPerQuantityPrimarySweepValues 10-80
GetRealDataValues 10-80

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

GetSweepNames ... 10-81
GetSweepUnits ... 10-82
GetSweepValues ... 10-83
IsDataComplex ... 10-84
IsPerQuantityPrimarySweep ... 10-84
Release Data ... 10-85
GroupPlotCurvesByGroupingStrategy ... 10-86
ImportIntoReport ... 10-87
ImportReportDataIntoReport ... 10-87
MovePlotCurvesToGroup ... 10-88
MovePlotCurvesToNewGroup ... 10-89
OpenWindowForAllReports ... 10-89
OpenWindowForReports ... 10-90
PastePlotSettings ... 10-90
PasteReports ... 10-91
PasteReportsWithLegacyNames ... 10-92
PasteTraces ... 10-92
PasteTracesWithLegacyNames ... 10-93
RenameReport ... 10-93
RenameTrace ... 10-94
ResetPlotSettings ... 10-95
SavePlotSettingsAsDefault ... 10-95
SetLinkOutputTraces ... 10-96
SetPropValue [Report Setup] ... 10-97
UnGroupPlotCurvesInGroup ... 10-97
UpdateAllReports ... 10-98
UpdateReports ... 10-98
UpdateTraces ... 10-99
UpdateTracesContextAndSweeps ... 10-101
11 - Boundary and Excitation 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.

General Commands Recognized by the Boundary Conditions Module 11-1
AddAssignmentTo 11-2
DeleteAllBoundaries 11-3
Delete Boundaries 11-3
Edit for BoundarySetup Commands 11-3
GetBoundaries 11-6
GetBoundariesOfType 11-6
GetAssignment for Boundaries or Excitations 11-7
GetDefaultBaseName 11-7
GetNumBoundaries 11-8
GetNumBoundariesOfType 11-8
Reassign a Boundary 11-8
RemoveAssignmentFrom 11-9
RenameBoundary 11-10
ReprioritizeBoundaries 11-10
SetDefaultBaseName 11-11
Icepak Boundary Condition Commands 11-11
AssignBlockBoundary 11-12
AssignGrilleBoundary 11-14
AssignNetworkBoundary 11-15
AssignOpeningBoundary 11-16
AssignRecircBoundary 11-17
AssignAdiabaticPlateBoundary 11-18
AssignConductingPlateBoundary 11-19
AssignResistanceBoundary 11-21
AssignSourceBoundary 11-24
AssignStationaryWallBoundary 11-25
AssignSymmetryWallBoundary 11-27
AssignEMLoss 11-27
AssignBlowerBoundary 11-29

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

AssignHeatExchangerBoundary 11-30
AssignSolarLoading 11-31
AssignInitialTemperature 11-32
12 - Native Component Module Script Commands 12-1
InsertNativeComponent (Fan) 12-1
InsertNativeComponent (Heatsink) 12-5
InsertNativeComponent (PCB) 12-8
InsertNativeComponent (CTM) 12-15
InsertNativeComponent 12-20
13 - Monitor Script Commands 13-1
AssignFaceMonitor 13-1
AssignPointMonitor 13-2
14 - Mesh Region Module Script Commands 14-1
Script Commands for Creating and Modifying Mesh Regions 14-1
EditGlobalMeshRegion 14-1
AssignMeshRegion 14-3
AssignMeshOperation 14-5
AssignMeshOperation (Mesh Sizes Region) 14-6
AssignVirtualMeshRegion 14-7
EditMeshRegion 14-9
15 - Analysis Setup Module Script Commands 15-1
ClearLinkedData (Module) 15-2
CopySetup 15-2
DeleteSetups 15-3
EditSetup 15-3
GetSetupCount 15-15
GetSetups 15-15
InsertSetup (Icepak - Steady State) 15-16
InsertSetup (Icepak - Transient) 15-24
PasteSetup 15-33

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

RenameSetup 15-33
RevertAllToInitial 15-33
RevertSetupToInitial 15-34
RevertSetupToInitialCondition 15-35
16 - Optimetrics Module Script Commands 16-1
CopySetup 16-6
DeleteSetups [Optimetrics] 16-7
DistributedAnalyzeSetup 16-7
EditSetup 16-8
EnableSetup 16-19
ExportDXConfigFile 16-20
ExportOptimetricsProfile 16-20
ExportOptimetricsResult 16-21
ExportParametricResults 16-22
ExportParametricSetupTable 16-23
ExportRespSurfaceMinMaxTable 16-23
ExportRespSurfaceRefinePoints 16-24
ExportRespSurfaceResponsePoints 16-25
ExportRespSurfaceVerificationPoints 16-25
GenerateVariationData [Parametric] 16-26
GetChildNames [Optimetrics] 16-26
GetChildObject [Optimetrics] 16-27
GetChildTypes [Optimetrics] 16-28
GetName 16-28
GetObjPath [Design] 16-29
GetOptimetricResult 16-29
GetOptimetricsResult 16-30
GetPropNames [Optimetrics] 16-31
GetPropValue [Optimetrics] 16-31
GetSetupNames [Optimetrics] 16-32

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

GetSetupNamesByType [Optimetrics] 16-33
ImportSetup 16-33
PasteSetup [Optimetrics] 16-34
RenameSetup [Optimetrics] 16-35
SetPropValue [Optimetrics] 16-35
SolveAllSetup 16-36
SolveSetup [Optimetrics] 16-37
General Commands Recognized by the Optimetrics Module 16-37
CopySetup 16-39
DeleteSetups [Optimetrics] 16-39
DistributedAnalyzeSetup 16-40
EditSetup 16-41
EnableSetup 16-52
ExportDXConfigFile 16-53
ExportOptimetricsProfile 16-53
ExportOptimetricsResult 16-54
ExportParametricResults 16-55
ExportParametricSetupTable 16-55
ExportRespSurfaceMinMaxTable 16-56
ExportRespSurfaceRefinePoints 16-57
ExportRespSurfaceResponsePoints 16-57
ExportRespSurfaceVerificationPoints 16-58
ExportDOEResponseCurve 16-59
ExportDOEResponseCurveSlices 16-59
ExportDOEResponseSurface 16-60
ExportDOELocalSensitivity 16-61
ExportDOELocalSensitivityCurve 16-61
GenerateVariationData [Parametric] 16-62
GetChildNames [Optimetrics] 16-63
GetChildObject [Optimetrics] 16-63

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

GetChildTypes [Optimetrics] ... 16-64
GetName ... 16-64
GetObjPath [Design] ... 16-65
GetOptimetricResult ... 16-65
GetPropNames [Optimetrics] ... 16-66
GetPropValue [Optimetrics] ... 16-67
GetSetupNames [Optimetrics] ... 16-67
GetSetupNamesByType [Optimetrics] ... 16-68
ImportSetup ... 16-69
PasteSetup [Optimetrics] ... 16-70
RenameSetup [Optimetrics] ... 16-70
SetPropValue [Optimetrics] ... 16-71
SolveAllSetup ... 16-71
SolveSetup [Optimetrics] ... 16-72
Parametric Script Commands ... 16-73
EditSetup [Parametric] ... 16-73
ExportParametricSetupTable ... 16-74
GenerateVariationData [Parametric] ... 16-74
InsertSetup [Parametric] ... 16-75
Optimization Script Commands ... 16-79
EditSetup [Optimization] ... 16-79
InsertSetup [Optimization] ... 16-83
Sensitivity Script Commands ... 16-88
EditSetup [Sensitivity] ... 16-88
InsertSetup [Sensitivity] ... 16-92
Statistical Script Commands ... 16-95
EditSetup [Statistical] ... 16-95
InsertSetup [Statistical] ... 16-95
17 - Solutions Module Script Commands ... 17-1
DeleteSolutionVariation ... 17-2

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

EditSources ... 17-3
ExportNMFData ... 17-3
ExportTransientData ... 17-4
FFTOnReport ... 17-5
GetAdaptiveSettings ... 17-6
GetAllSourceMagnitudes ... 17-6
GetAllSourceModes ... 17-7
GetAllSourcePhases ... 17-7
GetAllSources ... 17-8
GetAntennaParameters ... 17-8
GetFaceMeshCentroidAndArea ... 17-9
GetFaceMeshIDAndArea ... 17-9
GetFieldType ... 17-10
GetIncludePortPostProcessing ... 17-10
GetMultipactionBreakdown ... 17-11
GetNetworkDataSolution ... 17-12
GetNetworkDataSolutionDefinition ... 17-12
GetNetworkPostprocSetup ... 17-13
GetSourceContexts ... 17-13
GetTerminalExcitationType ... 17-14
GetTransientSolveTimes ... 17-14
SetSourceContexts ... 17-15
TDROnReport ... 17-16
18 - Field Overlays Module Script Commands ... 18-1
AddMarker[Fields Reporter] ... 18-1
AddMarkerToPlot ... 18-2
AddNamedExpression ... 18-3
CalcOp ... 18-3
CalcStack ... 18-4
CalculatorRead ... 18-5

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

CalculatorWrite 18-5
ChangeGeomSettings 18-6
ClcEval 18-7
ClcMaterial 18-7
ClcMaterialValue 18-8
ClearAllMarkers[Fields Reporter] 18-8
ClearAllNamedExpr 18-9
CopyNamedExprToStack 18-9
CreateFieldPlot 18-10
DeleteFieldPlot 18-13
DeleteMarker[Fields Reporter] 18-14
DeleteNamedExpr 18-15
DeleteUneditablePlot 18-15
DoesNamedExpressionExists 18-16
EditSurfaceMeshSummaryData 18-16
EnterComplex 18-18
EnterComplexVector 18-19
EnterCoord 18-20
EnterEdge 18-20
EnterLine 18-21
EnterOutputVar 18-21
EnterPoint 18-22
EnterQty 18-23
EnterScalar 18-23
EnterScalarFunc 18-24
EnterSurf 18-24
EnterVector 18-25
EnterVectorFunc 18-25
EnterVol 18-26
ExportFieldPlot 18-27

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

ExportMarkerTable 18-27
ExportOnGrid [Field Overlays] 18-28
ExportPlotImageToFile 18-30
ExportPlotImageWithViewToFile [Reporter] 18-31
ExportSurfaceMeshSummary 18-31
ExportToFile 18-33
GetFieldFolderNames 18-34
GetFieldPlotNames 18-34
GetFieldPlotQuantityName 18-35
GetMeshPlotNames 18-35
GetTopEntryValue 18-36
HidePolarPlot 18-37
HideRadiatedPlotOverlay 18-37
LoadNamedExpressions 18-38
ModifyFieldPlot 18-39
ReassignFieldPlot 18-40
RenameFieldPlot 18-42
RenamePlotFolder 18-43
SaveFieldsPlots 18-43
SaveNamedExpressions 18-44
SetFieldPlotSettings 18-45
SetPlotFolderSettings 18-46
UpdateAllFieldsPlots 18-50
UpdateQuantityFieldsPlots 18-50
19 - Fields Calculator Script Commands 19-1
AddNamedExpression 19-2
CalcOp 19-3
CalcRead(deprecated) 19-3
CalculatorRead 19-4
CalcStack 19-5

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

CalculatorWrite 19-5
ChangeGeomSettings 19-6
ClcEval 19-7
ClcMaterial 19-7
ClcMaterialValue 19-8
ClearAllNamedExpr 19-9
CopyNamedExprToStack 19-9
DeleteNamedExpr 19-10
DoesNamedExpressionExists 19-10
EnterComplex 19-11
EnterComplexVector 19-11
EnterCoord 19-12
EnterEdge 19-13
EnterLine 19-13
EnterOutputVar 19-14
EnterPoint 19-14
EnterQty 19-15
EnterScalar 19-16
EnterScalarFunc 19-16
EnterSurf 19-17
EnterVector 19-17
EnterVectorFunc 19-18
EnterVol 19-18
ExportOnGrid [Fields Calculator] 19-19
ExportToFile [Fields Calculator] 19-21
GetFieldsCalculatorExpressions 19-23
GetTopEntryValue 19-24
LoadNamedExpressions 19-24
SaveNamedExpressions 19-25
20 - Fields Summary Script Commands 20-1

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

EditFieldsSummarySetting 20-1
ExportFieldsSummary 20-2
21 - User-Defined Document Script Commands 21-1
AddDocument 21-1
DeleteAllDocuments 21-2
DeleteDocument 21-3
EditDocument 21-3
GetDocumentDefinitionNames 21-5
GetDocumentNames 21-5
RenameDocument 21-6
SaveHtmlDocumentAs 21-6
SavePdfDocumentAs 21-7
UpdateAllDocuments 21-8
UpdateDocument 21-8
ViewHtmlDocument 21-9
ViewPdfDocument 21-9
Example Python Script: Defining a Document 21-10
22 - User-Defined Solutions Commands 22-1
CreateUserDefinedSolution 22-1
DeleteUserDefinedSolutions 22-2
EditUserDefinedSolution 22-3
GetUserDefinedSolutionNames 22-4
GetUserDefinedSolutionProperties 22-4
23 - Network Data Explorer Script Commands 23-1
AddDiffPair 23-3
Cascade (SPISim) 23-4
ClearDiffPairs 23-5
Clone 23-6
Close 23-7
Combine (SPISim) 23-8

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

Deembed (SPISim) 23-9
DeembedBack (SPISim) 23-10
DeembedFront (SPISim) 23-11
DisableDiffPairs 23-12
EnableDiffPairs 23-13
ExportCitiFile 23-13
ExportFullWaveSpice 23-15
ExportMatlab 23-19
ExportNMFData 23-20
ExportNetworkData 23-20
ExportSpreadsheet 23-22
ExportTouchstone 23-24
ExportTouchstone2 23-27
Extract (SPISim) 23-29
GetFrequencies 23-30
GetFrequencyCount 23-30
GetName 23-31
GetPortCount 23-32
GetPortNumber 23-33
GetPostProcSettings 23-34
GetSolutionVariation 23-34
GetVariation 23-35
HasSameData 23-36
LoadSolution 23-38
Open 23-38
Rename (SPISim) 23-39
Renormalize (SPISim) 23-40
Reorder 23-41
Reorder (SPISim) 23-42
Reset 23-43

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

SetAllPortImpedances 23-44
SetPortDeembedDistance 23-45
SetPortImpedance 23-46
SetPostProcSettings 23-47
Smooth 23-48
Stretch (SPISim) 23-49
Terminate 23-50
24 - Definition Manager Script Commands 24-1
Add [component manager] 24-2
AddDataset 24-5
AddDefinitionFromBlock 24-7
AddMaterial 24-9
Add [padstack manager] 24-13
Add [symbol manager] 24-19
DeleteDataset 24-27
Edit [component manager] 24-27
EditDataset 24-40
EditMaterial 24-42
Edit [padstack manager] 24-48
ExportDataset 24-54
Export [footprint manager] 24-54
ExportMaterial 24-55
Export [padstack manager] 24-56
ExportScript 24-57
Export [symbol manager] 24-57
GetProjectMaterialNames 24-58
GetPropertyValue 24-59
ImportDataset 24-60
Note About File Types: 24-61
Remove [component manager] 24-62

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

Remove [footprint manager] 24-63
RemoveMaterial 24-64
Remove [padstack manager] 24-65
RemoveScript 24-65
Remove [symbol manager] 24-66
RemoveUnusedDefinitions 24-67
SetPropertyValue 24-68
UpdateDefFromBlock 24-69
Material Manager Script Commands 24-71
GetNames [material manager] 24-71
GetProperties [material manager] 24-72
IsUsed [material manager] 24-73
RemoveUnused [material manager] 24-73
Model Manager Script Commands 24-74
Add [model manager] 24-74
ConvertToDynamic 24-80
ConvertToParametric 24-81
Edit [deprecated] 24-81
EditWithComps [model manager] 24-81
Export [model manager] 24-88
GetNames [model manager] 24-88
IsUsed [model manager] 24-89
Remove [model manager] 24-89
RemoveUnused [model manager] 24-90
Script and Library Scripts 24-91
AddScript 24-91
EditScript 24-92
ExportScript 24-93
ModifyLibraries 24-94
RemoveScript 24-95

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

Symbol Manager Script Commands 24-96
Add [symbol manager] 24-96
BringToFront [symbol manager] 24-105
Edit [deprecated] 24-105
EditSymbolAndUpdateComps [symbol manager] 24-105
Export [symbol manager] 24-114
GetNames [symbol manager] 24-115
IsUsed [symbol manager] 24-115
Remove [symbol manager] 24-116
RemoveUnused [symbol manager] 24-117
25 - Core Global Script Context Commands 25-1
AddErrorMessage 25-1
AddFatalMessage 25-1
AddInfoMessage 25-2
AddWarningMessage 25-2
LogDebug 25-3
LogError 25-3
26 - Example Scripts 26-1
IronPython Example Scripts 26-1
BH Coordinates Python Script 26-1
Script Contents 26-2
Posco_BH_Curve.tab Contents 26-3
Equation Based Curve Python Script 26-6
HFSS Waveguide Array Python Script 26-9
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.

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

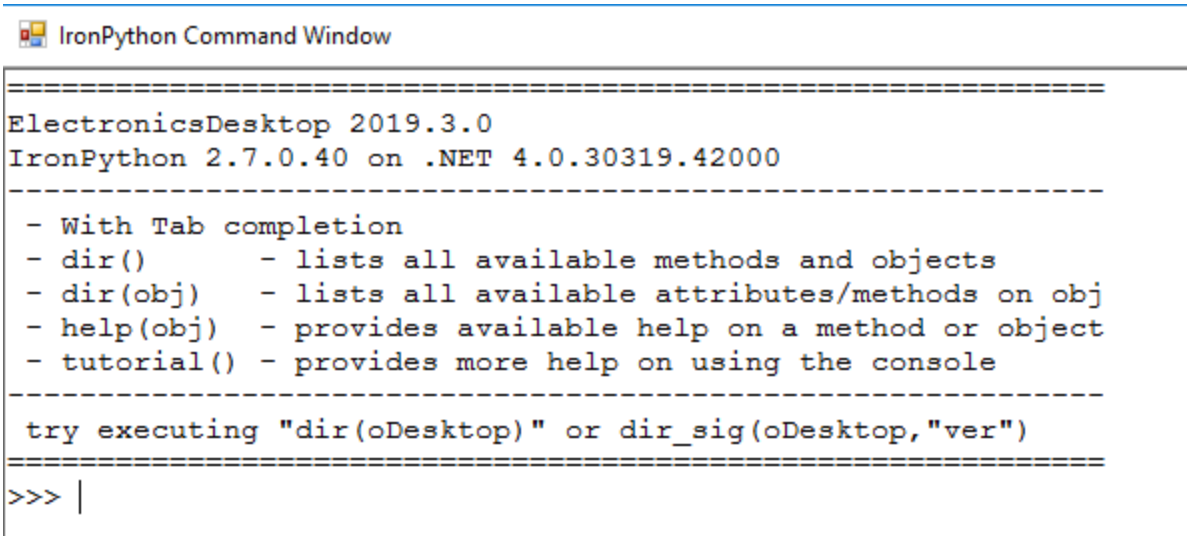
The advantages that IronPython use provides are significant:

- 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

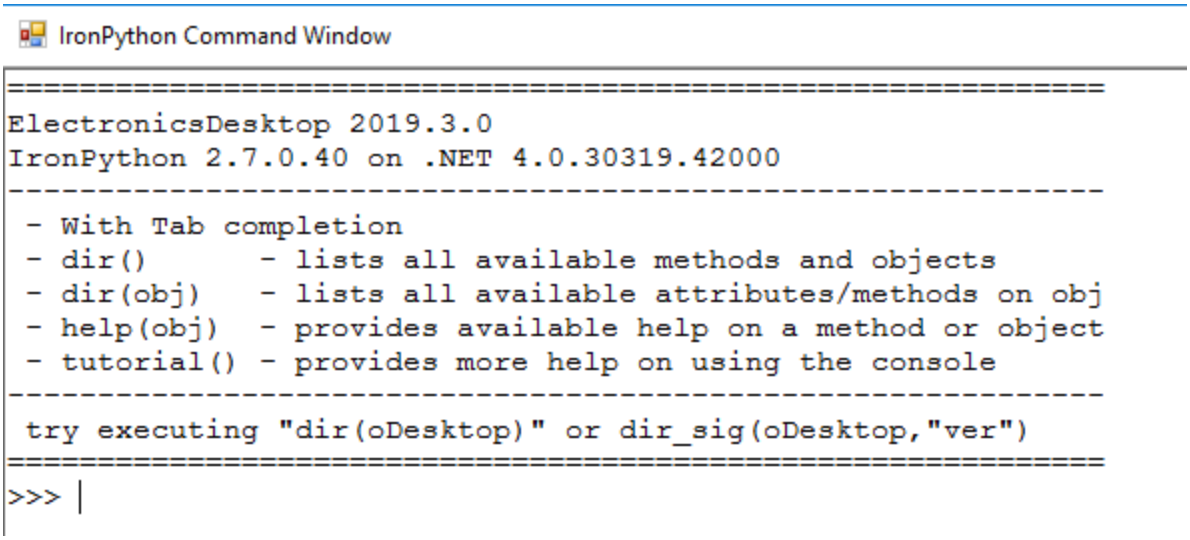
Appendix: IronPython Samples

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:**



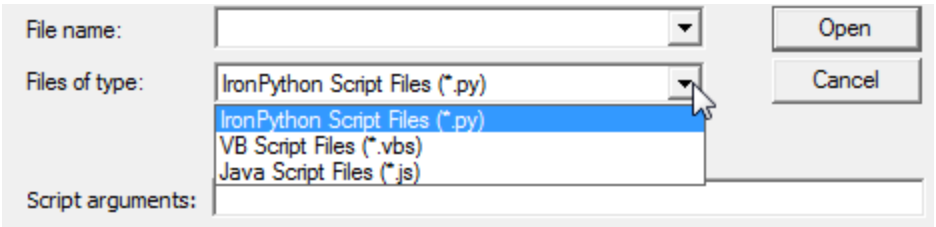
- **Tools** > **Run Script** menu item, select "IronPython" from the file type drop-down list.
- Launch the product with a script argument.
- Register an IronPython script as an external tool using the **Tools** > **External Tools** menu item.

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", ""

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", _
```


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

```
 "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.

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.

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

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.

- `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)`

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

`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",

  "IsCovered:= ", True,

  "XStart:= ", "-0.2mm",

  "YStart:= ", "-3mm",

  "ZStart:= ", "0mm",

  "Width:= ", "0.8mm",

  "Height:= ", "1.2mm",

  "WhichAxis:= ", "Z"

],
```


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:= ", "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.

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(
```


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: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
   }]]
 ])
```

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()
```


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.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" : "",
   "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()
```


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.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",
   "Name" : firstConeName,
   "Flags" : "",
   "Color" : "(132 132 193)",
   "Transparency" : 0,
   "PartCoordinateSystem": "Global",
   "UDMId" : "",
   "MaterialValue" : "\"vacuum\"",
   "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.

```
    )

# 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
   }
 )

 # 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(
   {
```


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" : 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).

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.

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

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):

- 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

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

'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 parameter	udmDefinition = udmFunctionLibrary.GetDefinition() (Note: callbackData is omitted in py interface)
long detachlFaces(int nFacesAndPartIds, long* faceIds, long* partIds, void* callbackData); where partIds is an output parameter	(bRet, partIds) = udmFunctionLibrary.DetachlFaces (nFacesAndPartIds, faceIds) (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.

'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, long* numFaces, long** faceIds, void* callbackData); where numFaces and faceIds are output parameters and numFaces is the size of faceId.	faceIds = udmFunctionLibrary.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);	lret = udpFunctionLibrary.Unite (objectIds)

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
where numObjects and objectIds are input parameters and numObjects is the size of 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)
```

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,	(lret, cloneIds) = udmFunctionLibrary.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.

C	Python
UDPCoordinateSystemAxis axis, double angle, int nClones, long* nClones, void* callbackData);	(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

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 *
```


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

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.

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

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

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()

- 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

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 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")

    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:

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

```
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)

- 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

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

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 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

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

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

Example UDM

```
class UDMExtension(IUDMExtension):


  def IsAttachedToExternalEditor(self):
    return False


  def GetInfo(self)
    udmInfo = UDMInfo(
      name = "SampleUDM",
      purpose = "udm example",
```


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

```
        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);


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)

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

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:

```
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**)**

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

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**)**
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**)**

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

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**)**
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**)**

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

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:

- **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,
```


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

```
    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:

```
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]
```


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

```
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, UnitType unitType, double defaultValue)	string Name string Description UnitType UnitType double DefaultValue
UDPParam	UDPParam() UDPParam(ParamDataType dataType, object data) object can be int, double, string, bool or UDPPosition	ParamDataType DataType object Data object can be int, double , string, bool or UDPPosition

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
	methods: setInt(int val) setBool(bool val) setString(string val) setDouble(double val) setPosition(UDPPosition val)	
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 z)	double X double Y 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
UDPPolylineSeg-mentDefinition	UDPPolylineSeg-mentDefinition(	PolylineSegmentType Seg-mentType

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
	PolylineSegmentType segmentType, int segmentStartIndex, int numberOfPoints, double angle, UDPPosition centerPoint, CoordinateSystemPlane arcPlane)	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, double tStart, double tEnd, int numOfPointsOnCurve)	string FunctionXt string FunctionYt string FunctionZt double TStart double TEnd int NumOfPointsOnCurve
UDPEquationBasedSurfaceDefinition	UDPEquationBasedSurfaceDefinition(string functionXuv, string functionYuv, string functionZuv, double uStart, double uEnd, double vStart,	string FunctionXuv string FunctionYuv string FunctionZuv double UStart double UEnd double VStart double VEnd

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 vEnd int reserved1 int reserved2)	two integer arguments that are reserved for future use. They need to be provided, for example as 0. For example: `theSurfaceDefinition = UDPEqua-tionBasedSur-faceDefinition ("u","v","1",0,1,0,1,0,0)`
UDPHelixDefinition	UDPHelixDefinition(int profileID, UDPPosition ptOnAxis, UDPPosition axisDir, double noOfTurns, bool isRightHanded, double radi-usChangePerTurn, 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, bool isRightHanded, double radi-usChangePerTurn)	int ProfileID UDPPosition PtOnAxis UDPPosition AxisDir double NoOfTurns bool IsRightHanded double RadiusChangePerTurn
UDPBLNDElements	UDPBLNDElements(int partID, int noOfEdges; int* listOfEdges;) UDPBLNDElements(int partID,	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;

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
	int noOfVertices; int* listOfVertices;)	int* listOfEdges; /* edges to be blended */ int noOfVertices; int* listOfVertices; /* vertices to be blended */
UDPBLNDFilletOptions	UDPBLNDFilletOptions(bool supressFillet, BLNDFilletRadiusLaw filletRadiusLaw, double filletStartRadius, double filletEndRadius, bool followSmoothEdgeSequence, BLNDFilletType filletType, double setbackDistance, double bulgeFactor)	bool SupressFillet /* Reserved for future */ BLNDFilletRadiusLaw FilletRadiusLaw double FilletStartRadius double FilletEndRadius bool FollowSmoothEdgeSequence /* Reserved for future */ BLNDFilletType FilletType double SetbackDistance double BulgeFactor /* Reserved for future */
UDPBLNDChamferOptions	UDPBLNDChamferOptions(bool supressChamfer, BLNDChamferRangeLaw chamferRangeLaw, double chamferLeftRange, double chamferRightRange)	bool SupressChamfer BLNDChamferRangeLaw ChamferRangeLaw double ChamferLeftRange 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 Name string Purpose string Company

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 company, string date, string version)	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 name, UDParam value, UnitType unitType, ParamPropType propType, ParamPropFlag propFlag)	string ParamName UDPParam ParamValue UnitType UnitType ParamPropType PropType ParamPropFlag PropFlag

UDP/UDM Constants

Full names of enum constants must be used in scripts.

Example:

```
unitType = UnitType.LengthUnit
```


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

```
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 ReadOnly MustBeInt MustBeReal Hidden Unknown
CoordinateSystemAxis	XAxis YAxis ZAxis
CoordinateSystemPlane	XYPlane YZPlane

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
	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.*/ 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

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

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(
    "y_size",
    "",
    UnitType.LengthUnit,
    ParamPropType.Value,
    ParamPropFlag.MustBeReal,
    UDPParam(ParamDataType.Double, 5)),
  UDPPrimitiveParameterDefinition2(
    "z_size",
    "",
    UnitType.LengthUnit,
    ParamPropType.Value,
```


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

```
    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)

    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],
```


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

```
    [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)


# 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
```


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

```
    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
    :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):
    """
```


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

```
    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:

- 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")
```


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

```
# 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

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.

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

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:

- **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:

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

- **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:

```
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

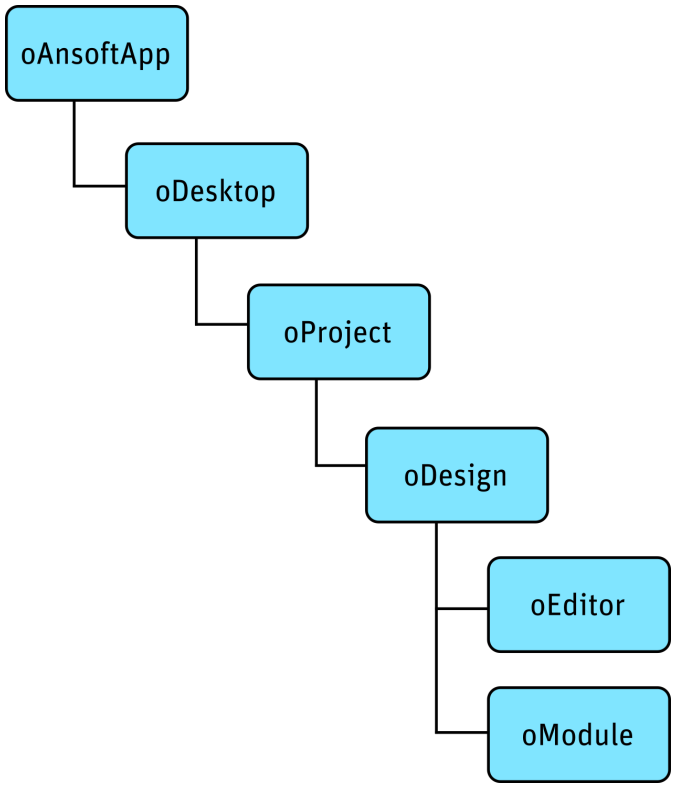
Working with Project Scripts

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

> **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()
```

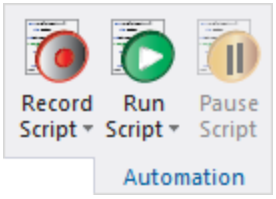
Running a Script

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

Within Electronics Desktop

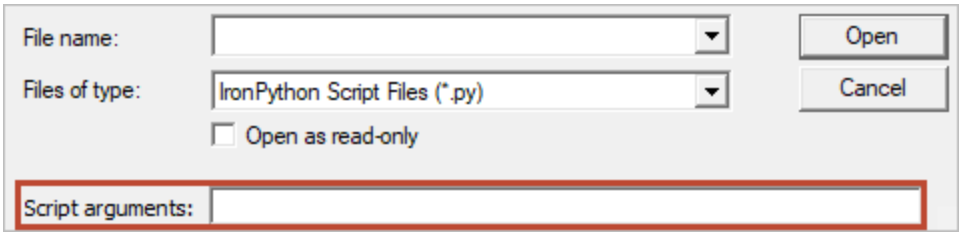
To run scripts in Electronics Desktop:

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 Icepak 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.

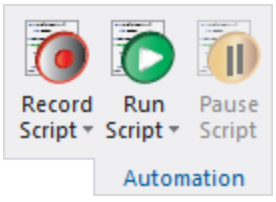
Scripts can be saved to an external file, or to the project.

> **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:



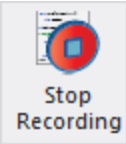
   A **Save As** file browser appears.

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.



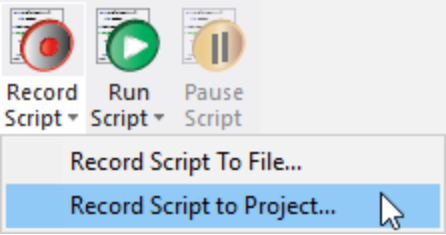
5. Perform the steps you want to record.
6. When you have finished recording the script, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

   The recorded script is saved to the folder you specified.

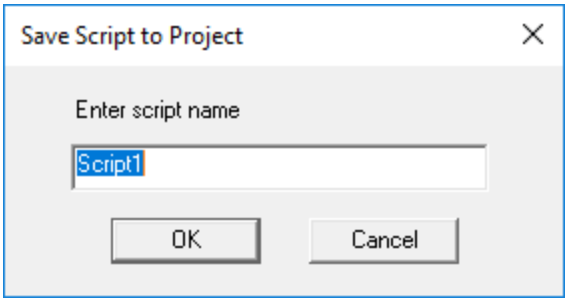
Recording a Script to a Project

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.

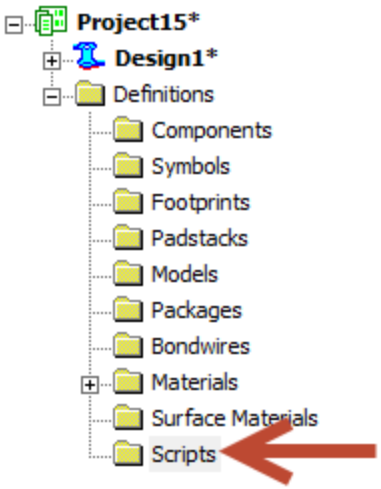
3. Perform the steps you want to record.
4. When you have finished, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

   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

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, ...
   ```

2. Named Array, where name precedes array.

   For example:

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


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

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"
    ],
  "Type:=", "EdgePort",
  "IsGapSource:=", true,
  "UpperProbe:=", false,
```


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

```
   "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 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.

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

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.

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.

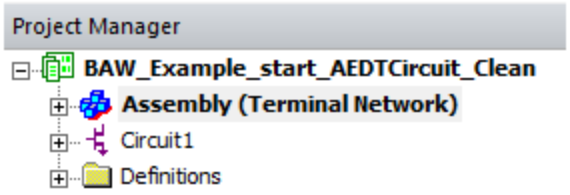
Basic Functions Overview

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:

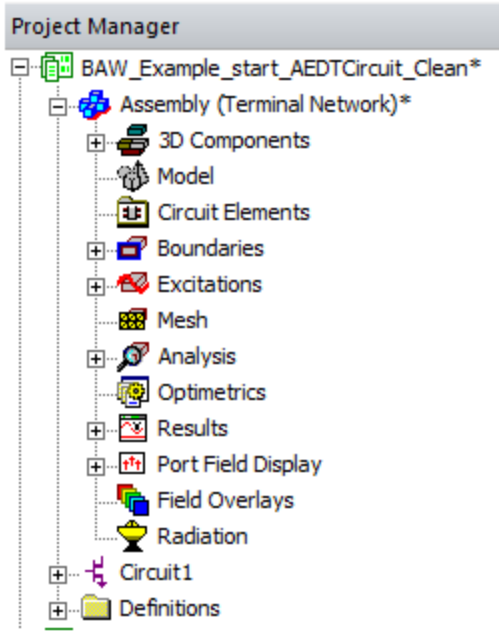
```
>>> oProject.GetChildNames()
['Assembly','Circuit1']
```

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.

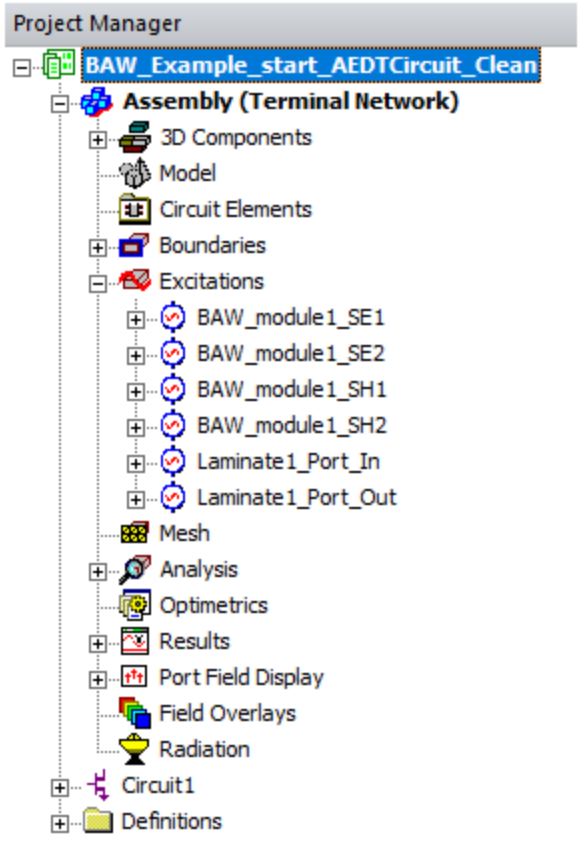


Any object in the **Project Manager** that has the '+' symbol for expansion is populated with child objects that can be queried. Once again, GetChildObject() can instance these objects. For example, set an instance for the Excitations:

```
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.



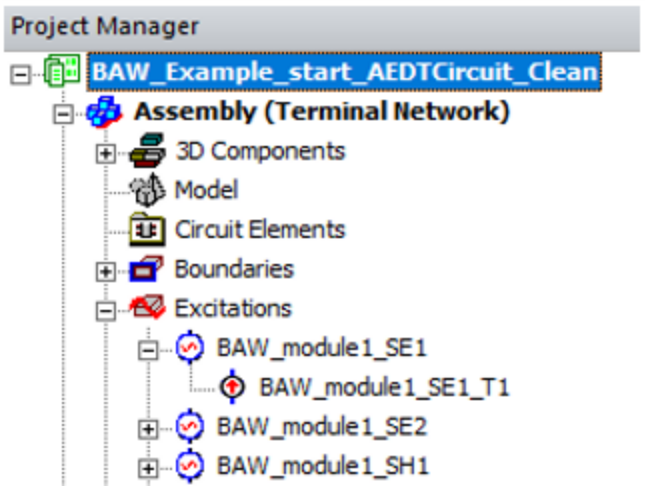
Those ports are outputted using GetChildNames():

```
>>> 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:



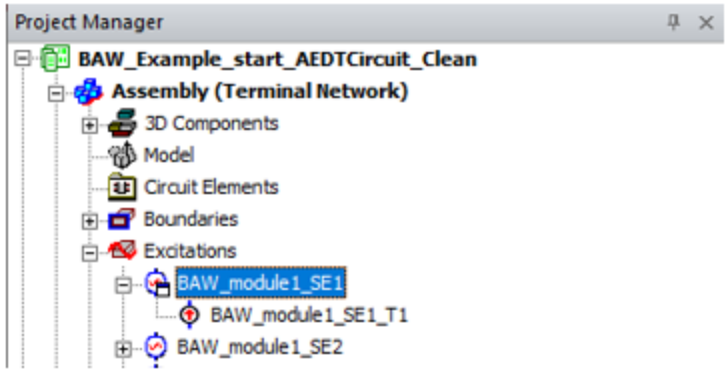
Through scripting, first define the Port Object (in this example, oPort) using the 'GetChildObject()' command for the first port, 'BAW_module1_SE1.' Then determine its Object Child Name, the terminal definition:

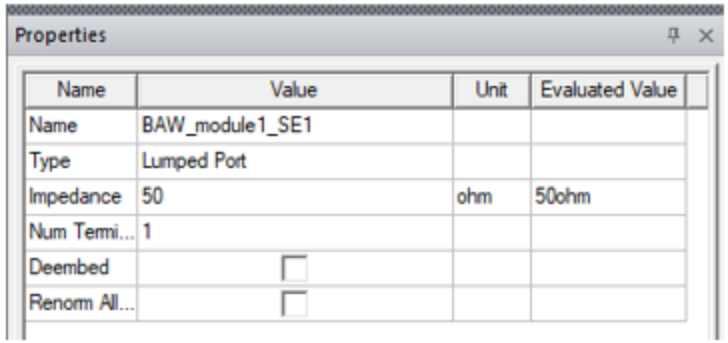
```
>>> oPort = oExcitations.GetChildObject('BAW_module1_SE1')
>>> oPort.GetChildNames()
['BAW_module1_SE1_T1']
```

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']
```

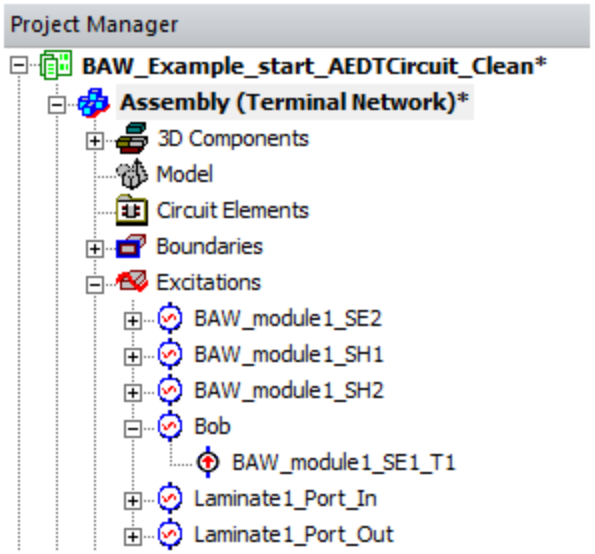
Once you identify the desired object and know the desired property, you can access the value via GetPropValue(). For example, if you want to retrieve the name of the object oPort:

```
>>> 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']
```

Additional Details Specific to Icepak Monitor and Mesh Modules

GetChildTypes of the Monitor module returns the two types of its children, which are Face and Point. These refer to the two types of monitors (i.e., face and point monitors that can be created by the user).

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("monitor").GetChildTypes()
['Face', 'Point']
```

GetChildTypes of the Mesh module returns the two types of its children, which are Operation and Region. These refer to the Mesh operations specified and the Mesh Regions created in the design.

```
>>> design.GetChildObject("mesh").GetChildTypes()
['Operation', 'Region']
```

3D Component Encapsulation

These script interfaces are compliant with encapsulation. For example:

- 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()
[]
```


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

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")
```

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.

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 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.

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

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

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

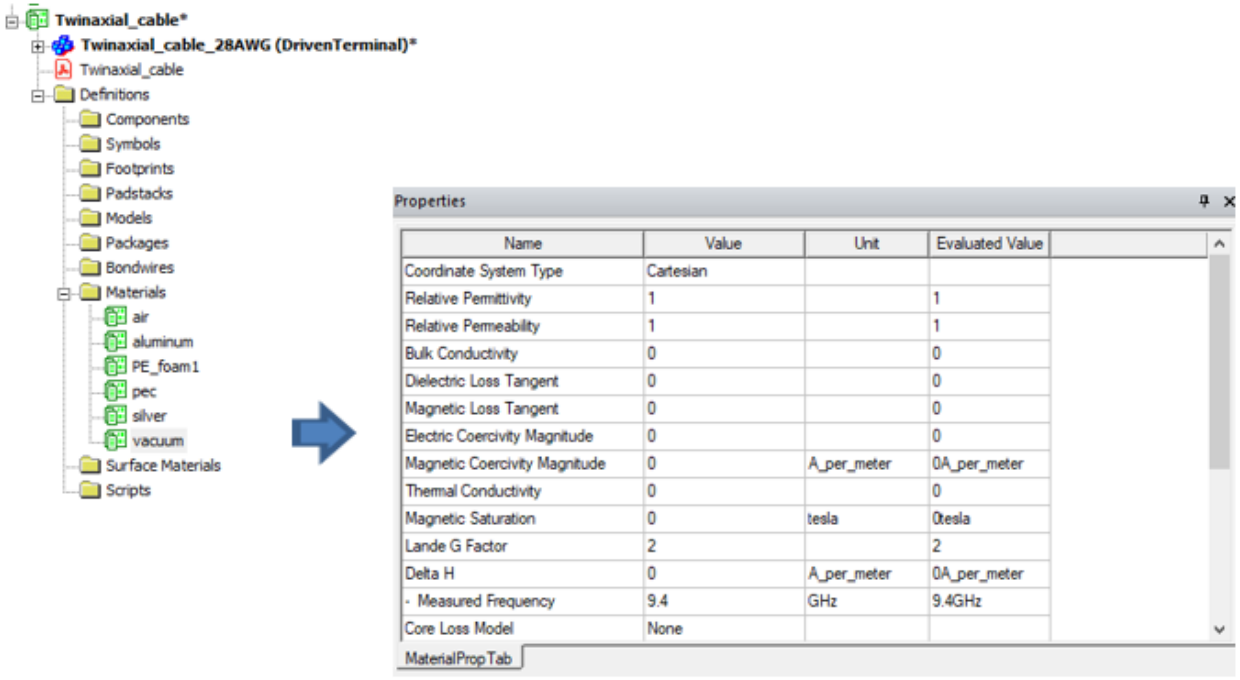
>>> omat.GetPropNames()

['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.

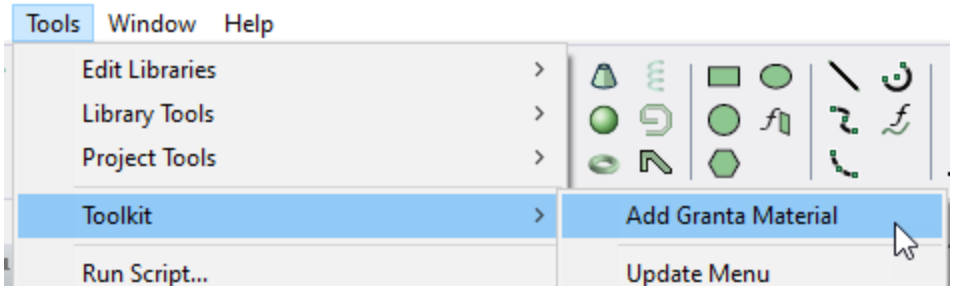
With this feature enabled you can then:

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

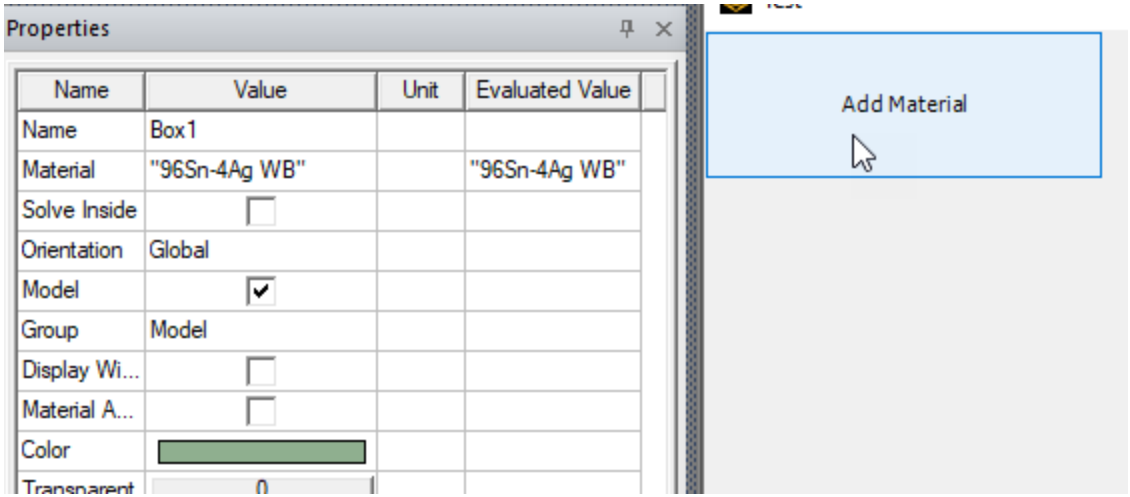
Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- 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

GetArrayVariables

GetProperties

GetPropertyValue

GetVariables

GetVariableValue

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

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 "/" .
- 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.
- 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.

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

- 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.
- 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"

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

- <*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.

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..

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

- 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.

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


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

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.
- 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.

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

```
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

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')

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

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
- 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

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

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

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.
- GetPropNames(bIncludeReadOnly) will return the property names listed in the property window when the setup 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.

- 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:

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

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

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"

For 3D Model Editor objects:

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

<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.

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

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 strings containing names of variables.

Python Syntax	GetArrayVariables()
Python Example	`oProject.GetArrayVariables()` `oDesign.GetArrayVariables()`

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") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables")

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

			• 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.

> **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")

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

			• 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:Inst@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")`
	`...`

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`.

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 variables.

Python Syntax	GetVariables ()
Python Example	`oProject.GetVariables()` `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")`

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") • Signals ("Signals")
	<propServer>	String	An object identifier, generally returned from another script method, such as CompInst@R;2;3
	<propName>	String	Name of the property.
	<propValue>	String	The value for 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	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.

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

Count

CreateComponent2_0

DeleteProject

DeleteRegistryEntry

DoesRegistryValueExist

DownloadJobResults

EditComponent2_0

EnableAutoSave

ExportOptionsFiles

GetActiveProject

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

GetActiveScheduler

GetActiveSchedulerInfo

GetAutoSaveEnabled

GetBuildDateTimeString

GetCustomMenuSet

GetDefaultUnit

GetDesktopConfiguration

GetDistributedAnalysisMachines

GetDistributedAnalysisMachinesForDesignType

GetExeDir

GetGDIObjectCount

GetLibraryDirectory

GetLocalizationHelper

GetMessages

GetMonitorData

GetPersonalLibDirectory

GetProcessID

GetProjectDirectory

GetProjectList

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

GetRegistryString

GetRunningInstancesMgr

GetSchematicEnvironment

GetScriptingToolsHelper

GetSysLibDirectory

GetTempDirectory

GetUserLibDirectory

GetVersion

InsertComponent2_0

IsFeatureEnabled

KeepDesktopResponsive

LaunchJobMonitor

NewProject

OpenAndConvertProject

OpenProject

OpenProjectWithConversion

PageSetup

PauseRecording

PauseScript

Print

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

QuitApplication

RefreshJobMonitor

ResetLogging

RestoreProjectArchive

RestoreWindow

ResumeRecording

RunACTWizardScript

RunProgram

RunScript

RunScriptWithArguments

SelectScheduler

SetActiveProject

SetActiveProjectByPath

SetCustomMenuSet

SetDesktopConfiguration

SetLibraryDirectory

SetProjectDirectory

SetRegistryFromFile

SetRegistryInt

SetRegistryString

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

SetSchematicEnvironment

SetTempDirectory

ShowDockingWindow

Sleep

StopSimulations

SubmitJob

TileWindows

Related Topics:

Desktop Commands For Registry Values

ImportExport Tool Commands

Component 2.0 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.

<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><projectName></td><td>String</td><td>Project name. Passing an empty string adds the message as the desktop global message.</td></tr>
<tr><td><designName></td><td>String</td><td>Design name. Ignored if project name is empty. Passing an empty string adds the message to project node in the message tree.</td></tr>
<tr><td><severity></td><td>Integer</td><td>One of "Error", "Warning" or "Info". Anything other than the first two is treated as "Info"

0 = Informational, 1 = Warning, 2 = Error, 3 = Fatal</td></tr>
<tr><td><msg></td><td>String</td><td>The message for the message window.</td></tr>
<tr><td><category></td><td>String</td><td>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.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	AddMessage(*<projectName>*, *<designName>*, *<severity>*, *<msg>*, *<category>*)
Python Example	`oDesktop.AddMessage("Project1", "IcepakDesign", 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.

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).

UI Access

In **Message Manager**, right-click and select **Clear messages for [*ProjectName*]...**

Parameters

Name	Type	Description
<projectName>	String	Name of the project from which to clear messages.
<designName>	String	Name of the design under the <projectName> from which to clear messages.
<startSeverity>	Integer	User-specified severity level at which messages start getting cleared: • 0 = informational • 1 = warning • 2 = error • 3 = fatal error
<stopSeverity>	Integer	*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.

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	ClearMessages(<*projectName*>, <*designName*>, <*startSeverity*>, <*stopSeverity*>)

Python Example	**Preserve Current Functionality; stopSeverity = 0 (DEFAULT VALUE)**
	`# startSeverity = 0; clears just the Info messages`
	`oDesktop.ClearMessages("Ex_EC_1_Asymmetrical_Conductor", "Maxwell3DDesign1", 0)`
	`# 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`

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

```
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)


# 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)
```


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

	# 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)
	
	# 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**.

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	CloseAllWindows()
Python Example	`oDesktop.CloseAllWindows()`

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")`

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

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 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

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	Count

Python Example	Example with three projects open in the Electronic Desktop:
	`projects = oDesktop.GetProjects()`
	`numprojects = projects.Count`
	`numprojects`
	`3`

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")`

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

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.

<table>
<tr><td>UI Access</td><td>Select Scheduler</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><jobID></td><td>String</td><td>Provide the job ID of the target job. The job ID must be able to be found in current .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.</td></tr>
<tr><td><projectPath></td><td>String</td><td>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.</td></tr>
<tr><td><resultPath></td><td>String</td><td>A string giving the folder path for the download to save to.</td></tr>
<tr><td><Filters> (optional)</td><td>String</td><td>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.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>A Boolean result about download complete or not</td></tr>
</table>

Python Syntax	DownloadJobResults(*jobID*, *projectPath*, *resultsPath*, *filters*)
Python Example	`boolDownloadCompleted = oDesktop.DownloadJobResults("01234567890123456789", "C:\\projects\\basic.aedt", "C:\\projects\\DownloadResults\\", "*")`

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

DeleteRegistryEntry

Deletes a registry entry from a registry key. Returns true if deletion succeeded.

UI Access	N/A
Parameters	Name: *<pathToRegistry>* — Type: String — Description: Path to Registry entry.
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: *<KeyName>* — Type: String — Description: Full name of registry key, including path.
Return Value	Boolean: • **True** – Key exists.

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

	• **False** – Key does not exist.

Python Syntax	DoesRegistryValueExist(<*KeyName*>)
Python Example	`Exist = oDesktop.DoesRegistryValueExist('Desktop/ActiveDSOConfigurations/Icepak')`

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)`

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

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		

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.

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

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()`

GetBuildDateTimeString

Returns a string representing the build date and time of the product.

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 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’.

Python Syntax	GetCustomMenuSet ()
Python Example	`oDesktop.GetCustomMenuSet()`

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

GetDefaultUnit

Returns the default unit for a physical quantity.

<table>
<tr><td>UI Access</td><td>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.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><type></td><td>String</td><td>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)
"Density"
"Flux"
"Force"
"Frequency"</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.

			• "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").		

Python Syntax	GetDefaultUnit(<type>)
Python Example	`oDesktop.GetDefaultUnit("Length")`

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

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.
Return Value	Returns a collection of names of machines used for distributed analysis.

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	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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><designTypeName></td><td>String</td><td>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"</td></tr></table>
Return Value	Object; returns a collection of machine names.

Python Syntax	GetDistributedAnalysisMachinesForDesignType (<designTypeName>)
Python Example	`machineNames =oDesktop.GetDistributedAnalysisMachinesForDesignType ("Icepak")`

GetExeDir

Returns the path where the executable is located.

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 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()`

GetLibraryDirectory

Get the path to the `SysLib` directory.

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	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 | Type | Description None | |
Return Value	Object Localization helper object, such as "IDispatch(ILocalizationHelper)"

Python Syntax	GetLocalizationHelper()

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.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)

Python Syntax	GetMessages (<*ProjectName*>, <*DesignName*>, <*Severity*>)
Python Example	`Messages = oDesktop.GetMessages("MyProject", "Icepak1", 1)`

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

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.

UI Access	N/A
Parameters	None.
Return Value	Integer process ID of ansysedt.exe. For example, 12716.

Python Syntax	GetProcessID ()

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.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.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the names of all open projects 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.

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()`

GetRegistryInt

Obtains registry key integer 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.

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/Icepak/UpdateReportsDynamicallyOnEdits')`

GetRegistryString

Obtains registry key string value.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | 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.

Python Syntax	GetRegistryString(<*KeyName*>)
Python Example	`activeDSO = oDesktop.GetRegistryString('Desktop/ActiveDSOConfigurations/Icepak')`

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

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()`

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

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	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
Parameters	Name | Type | Description None | |
Return Value	Object ScriptingTools helper object

Python Syntax	GetScriptingToolsHelper()
Python Example	`oDesktop.GetScriptingToolsHelper()`

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

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()`

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()`

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

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()`

GetVersion

Returns a string representing the version.

UI Access	N/A
Parameters	None.
Return Value	String containing version of the product.

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	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.

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",_`

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

```
 "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.

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)`

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

LaunchJobMonitor

Starts job monitoring. This brings up the **Monitor Job** dialog box.

UI Access	**Tools** > **Job Management** > **Monitor Jobs**		
Parameters	Name	Type	Description
	<projectPath>	String	Path to the project file to be monitored
Return Value	None		

Python Syntax	LaunchJobMonitor(*<projectPath>*)
Python Example	`oDesktop.LaunchJobMonitor("C:\\projects\\basic.aedt")`

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()`

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

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>*)
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()`

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.

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	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**.

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><*ArchiveFilePath*></td><td>String</td><td>Path to archived file</td></tr><tr><td><*ProjectFilePath*></td><td>String</td><td>Path to restore location</td></tr><tr><td><*OverwriteExistingFiles*></td><td>Boolean</td><td>True to overwrite an existing file of the same name; else False.</td></tr><tr><td><*OpenProjectAfterRestore*></td><td>Boolean</td><td>True to open the project after it is restored; else False.</td></tr></table>
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 `None`</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><tr><td><*SCriptPath*></td><td>String</td><td>Name or full path of the script to execute. If the full path to the script is not specified, Twin Builder searches for the spe-</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.

			cified 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><th>Name</th><th>Type</th><th>Description</th></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>

Python Syntax	RunScriptWithArguments (<*ScriptPath*>, <*Arguments*>)

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.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
	<configName>	String	Name of desired Desktop configuration. Can be one 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.

			• '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*>)
Python Example	`oDesktop.SetLibraryDirectory("c:\libraries")`

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

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>
Return Value	Success if configuration is imported. Bad argument value if the file is not found or does not contain valid analysis configuration 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	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/Icepak/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	<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><tr><td><*value*></td><td>String</td><td>String value to be assigned to registry key.</td></tr></table>
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/Icepak', '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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*schEnv*></td><td>Integer</td><td>Desired schematic environment. Can be one of: 0 (Circuit) 1 (Twin Builder) </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.

	• 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>
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	Sleep (<*TimeInMilliseconds*>)
Python Example	`oDesktop.Sleep(1000)`

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"))`

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

	moreIDs = oDesktop.SubmitJob("C:\\hpc-settings\\Submit_ Job_Settings.areg", "C:\\projects\\basic.aedt", "Design1", "Setup1") With an RSM scheduler with LSDSO: 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)

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

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 Icepak.
2. Go to the DSO and HPC options and create a setup named "test".
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/Icepak", "test")
```

See the following subtopics:

DeleteRegistryEntry

DoesRegistryValueExist

GetRegistryInt

GetRegistryString

SetRegistryFromFile

SetRegistryInt

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

SetRegistryString

DeleteRegistryEntry

Deletes a registry entry from a registry key. Returns true if deletion succeeded.

UI Access	N/A
Parameters	Name: *<pathToRegistry>* — Type: String — Description: Path to Registry entry.
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: *<KeyName>* — Type: String — Description: Full name of registry key, including path.
Return Value	Boolean:

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

	• **True** – Key exists. • **False** – Key does not exist.

Python Syntax	DoesRegistryValueExist(<*KeyName*>)
Python Example	`Exist = oDesktop.DoesRegistryValueExist('Desktop/ActiveDSOConfigurations/Icepak')`

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/Icepak/UpdateReportsDynamicallyOnEdits')`

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

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.

Python Syntax	GetRegistryString(*<KeyName>*)
Python Example	`activeDSO = oDesktop.GetRegistryString('Desktop/ActiveDSOConfigurations/Icepak')`

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	Name: *<filePath>* — Type: String — Description: Full file path of registry file.
Return Value	Success if configuration is imported. Bad argument value if the file is not found or does not contain valid analysis configuration 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	SetRegistryFromFile(<*filePath*>)
Python Example	`oDesktop.SetRegistryFromFile('c:/temp/test.acf')`

SetRegistryInt

Sets registry key to an integer value.

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><tr><td><*int*></td><td>Integer</td><td>Integer value to be assigned to registry key.</td></tr></table>
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/Icepak/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/Icepak', '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

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

ImportEDB

ImportExtracta

ImportGDSII

ImportGerber

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')`

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

	`oTool.ImportANF('C:\\AnsTranslator\\results\\package4.anf')`

ImportANFv2

Imports an ANFv2 file into a new project. For newer ANF files, use ImportANF.

UI Access	**File** > **Import** > **ANF**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ANFfilename*></td><td>String</td><td>Full path of ANF file.</td></tr><tr><td><*outputPathName*></td><td>String</td><td>Full path of *.aedb output file.</td></tr><tr><td><*controlFileName*></td><td>String</td><td>Full path of XML control file.</td></tr><tr><td><*cmpFileName*></td><td>String</td><td>Full path of CMP file. Pass empty string if none.</td></tr></table>
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", "")`

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

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.
<*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.

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

	<logFileName>	String	Full path of log file.
Return Value	None.		

Python Syntax	ImportAWRMicrowaveOffice (<*xmlFileName*>, <*outputPathName*>, <*logFileName*>)

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()`

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

	`oTool = oDesktop.GetTool('ImportExport')` `oTool.ImportEDB('C:/MyFiles/projectforimport.aedb/edb.def')`

ImportExtracta

Imports a Cadence Extracta file into a new project.

> **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', '')`

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	<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')`

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

ImportGerber

Imports a GERBER file into a new project.

UI Access	**File** > **Import** > **GERBER**.		
Parameters	Name	Type	Description
	<gerberFileName>	String	Full path of GERBER 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	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

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	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
<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.

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	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*>)

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",`

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

```
                "SubstrateMaterial:="   , "FR-4",
                "CreateBoard:="         , True,
                "ModelBoardAsRect:="    , True,
                "ModelDeviceAsRect:="   , False,
                "Cutoff:="              , False,
                "IncludeDrilledHoles:=" , True,
                "HoleDiameterCutoff:="  , "0.1mm",
                "ReplaceDevices:="      , False,
                "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	

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.

Return Value	None.

Python Syntax	ImportIDFandMerge(*<idfFileName>*, *<libPathName>*, *<controlFileName>*, *<project>*, *<design>*)
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.

	oTool = oDesktop.GetTool('ImportExport') oTool.ImportIDFandMerge('C:/Files/test.brd', 'C:/Files/test.aedb.lib', 'C:/Files/test.xml', 'Project1', 'Design12')

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

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

	<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
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",`

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

```
                "NumInternalLayer:="    , 2,
                "HighSurfaceCopper:="   , 30,
                "LowSurfaceCopper:="    , 30,
                "InternalLayerCopper:=" , 30,
                "TraceMaterial:="       , "Cu-Pure",
                "SubstrateMaterial:="   , "FR-4",
                "CreateBoard:="         , True,
                "ModelBoardAsRect:="    , False,
                "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*>)

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.ImportIPC('C:/Files/test.cvg', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml', 'C:/Files/test.txt')`

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')`

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

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.
	<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')`

ImportECXML

Imports a model in the EC XML format.

UI Access	**Icepak** > **Import ECXML**

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 *<ImportECXML>* | string | Full path of EC XMLfile.
Return Value	None

Python Syntax	ImportECXML (*<Full path to EC XML file>*)
Python Example	`oDesign.ImportECXML("C:\\Ansys\ECXML\Block.ecxml")`

ImportJEDEC

Imports a model in a JEDEC file format.

UI Access	**Icepak** > **Import JEDEC PTD/JEP30 File**
Parameters	Name | Type | Description *<ImportJEDEC>* | string | Full path of EC XMLfile.
Return Value	None

Python Syntax	ImportJEDEC (*<Full path to JEDEC file>*)
Python Example	`oDesign.ImportJEDEC("C:\\Ansys\JEDEC\Network.jptd")`

Component 2.0 (Beta) Commands

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

A 2.0 component is a project-level definition that can include one or more 3D designs. The component can be thought of as a library that can include 3D designs of different geometries and 3D designs that use different solvers. The individual designs in the 2.0 component are referred to as representations. The following commands are available:

CreateComponent2_0

EditComponent2_0

InsertComponent2_0

CreateComponent2_0

This command is used create the 2.0 component, which entails creating the first representation and assigning security to the entire component (as opposed to encrypting the individual representation).

Note that this topic includes two examples of how to use the CreateComponent2_0 command: the first example is a typical script that would be generated from the **Tools** > **Record Script to File** user interface function. The second example is a simplified version of the CreateComponent2_0 command that includes only the required script functions needed to create the component; this version can aid in automation.

> **Note:**
>
> A recorded script does not record Component 2.0 passwords; they must be added after the script is created.

<table>
<tr><td>UI Access</td><td>Right click on a design in the Project Manager tree and select Create Component 2.0</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ComponentData></td><td>Array</td><td>Structured array that defines the properties of the top-level 2.0 component. It has the following parameters:
<code>["NAME:ComponentData",</code>
<code>[ComponentHeaderInfoData],</code># Array that sets the information for the component file; see example below.</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.

			`[ComponentSecurityData]`, # Optional array that defines the overall security for the 2.0 component; see example below. `]`

	<CreateRepresentations Data>	Array	Structured array that defines the representation in the 2.0 component. There can be multiple <CreateRepresentationsData> definitions. It has the following parameters:
			`["NAME:CreateRepresentationsData",`
			`["NAME:RepresentationData",`
			`[HeaderAndGeometryData]` # Array that defines the representation parameters defined on the **Info**, **Model**, **Coordinate Systems**, **Parameters**, and **Encryption** tabs of the Create Component 2.0 Representation window; see example below.
			`[DesignData]` # Array that defines parameters on the design-specific tabs; which depending on the design may be **Boundaries**, **Excitations**, **Circuit Elements**, **Hybrid Regions**, and **Mesh** tabs of the Create Component 2.0 Representation window; see example below.
			`[ImageFile]` # Array that defines the image to display in the upper right area of the 3D modeler whenever the representation is used. The filename is a string that includes the filepath and file name. It can be left blank.
			`]`
			`]`

	<Filename>	String	Filepath of the Component 2.0 file to be created; it has the .acmp2 file extension.
Return Value	None		

Python Syntax	CreateComponent2_0(<*ComponentData*>, <*CreateRepresentationsData*>, <*Filename*>)
Python Example	The example below demonstrates the creation of the 2.0 component and the addition of two rep-

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

resentations.

```
oDesktop.OpenProject("D:/Designs/Waveguide_Filter.aedt")
oDesktop.CreateComponent2_0(
  [
    "NAME:ComponentData",
    [
      "NAME:ComponentHeaderInfoData",
      "ComponentName:=" , "Filter1",
      "Company:=" , "",
      "CompanyURL:=" , "",
      "ModelNumber:=" , "",
      "Version:=" , "1.0",
      "Notes:=" , "",
      "OwnerName:=" , "username",
      "OwnerEmail:=" , "user.name@company.com",
      "Date:=" , "9:12:32 AM Nov 17, 2025"
    ],
    [
      "NAME:ComponentSecurityData",
      "IsEditAllowed:=" , True,
      "NewEditPassword:=" , "111111"
    ]
  ],
  [
    "NAME:CreateRepresentationsData",
    [
      "NAME:RepresentationData",
```


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

```
[
  "NAME:HeaderAndGeometryData",
  "SourceDesignName:=" , "Waveguide_Filter:IcepakDesign1",
  "RepresentationName:=" , "IcepakDesign1",
  "Company:=" , "",
  "Company URL:=" , "",
  "Model Number:=" , "",
  "Help URL:=" , "",
  "Version:=" , "1.0",
  "Notes:=" , "",
  "IconType:=" , "",
  "Owner:=" , "username",
  "Email:=" , "user.name@company.com",
  "Date:=" , "9:12:35 AM Oct 27, 2025",
  "HasLabel:=" , False,
  "IsEncrypted:=" , True,
  "AllowEdit:=" , True,
  "SecurityMessage:=" , "",
  "Password:=" , "",
  "EditPassword:=" , "111111",
  "PasswordType:=" , "InternalPassword",
  "HideContents:=" , True,
  "ReplaceNames:=" , True,
  "ComponentOutline:=" , "None",
  "IncludedParts:=" , ["wg_flange_in","wg_inner","wg_outer","wg_
  flange_out"],
  "VisibleParts:=" , [],
  "IncludedCS:=" , [],
  "DefaultHandle:=" , "Global",
  "IncludedParameters:=" , [],
```


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

```
        "IncludedDependentParameters:=", [],
        "ParameterDescription:=", [],
        "IsLicensed:=" , False,
        "LicensingDllName:=" , "",
        "VendorComponentIdentifier:=", "",
        "PublicKeyFile:=" , ""
      ],
      [
        "NAME:DesignData"
      ],
      [
        "NAME:ImageFile",
        "ImageFile:=" , ""
      ]
    ],
    [
      "NAME:RepresentationData",
      [
        "NAME:HeaderAndGeometryData",
        "SourceDesignName:=" , "Waveguide_Filter:filter",
        "RepresentationName:=" , "filter",
        "Company:=" , "",
        "Company URL:=" , "",
        "Model Number:=" , "",
        "Help URL:=" , "",
        "Version:=" , "1.0",
        "Notes:=" , "",
        "IconType:=" , "",
```


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

```
"Owner:=" , "username",
"Email:=" , "user.name@company.com",
"Date:=" , "9:12:31 AM Oct 27, 2025",
"HasLabel:=" , False,
"IsEncrypted:=" , False,
"AllowEdit:=" , False,
"SecurityMessage:=" , "",
"Password:=" , "",
"EditPassword:=" , "",
"PasswordType:=" , "UnknownPassword",
"HideContents:=" , True,
"ReplaceNames:=" , True,
"ComponentOutline:=" , "None",
"IncludedParts:=" , ["wg_flange_in","wg_inner","wg_outer","wg_
flange_out"],
"VisibleParts:=" , [],
"IncludedCS:=" , [],
"DefaultHandle:=" , "Global",
"IncludedParameters:=" , ["fillet_radius","fillet_radius_
flange","len_wg1","len_wg2","len_wg3","t1","t2","t_
flange","w1","w2","w_flange","wg_height","wg_thickness","wg_width"],
"IncludedDependentParameters:=", [],
"ParameterDescription:=", [ "fillet_radius:=" , "", "fillet_radius_
flange:=", "", "len_wg1:=" , "", "len_wg2:=" , "", "len_wg3:=" , "",
 "t1:=" , "", "t2:=" , "", "t_flange:=" , "", "w1:=" , "", "w2:=" ,
"", "w_flange:=" , "", "wg_height:=" , "", "wg_thickness:=" , "",
"wg_width:=" , ""],
"IsLicensed:=" , False,
"LicensingDllName:=" , "",
"VendorComponentIdentifier:=", "",
"PublicKeyFile:=" , ""
```


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",
        "Excitations:=" , ["1","2"]
      ],
      [
        "NAME:ImageFile",
        "ImageFile:=" , ""
      ]
    ]
  ],
  "D:/Designs/AAProjects/ComponentCmd/test.acmp2")
```

Python Example

The example below is a simplified version of the CreateCommand2_0 command; it only includes the necessary functions, but users can add other functions (as shown above) as desired.

```
oDesktop.CreateComponent2_0(
  [
    "NAME:ComponentData",
    [
      "NAME:ComponentHeaderInfoData",
      "ComponentName:=" , "Test1",
      "Company:=" , "",
      "CompanyURL:=" , "",
      "ModelNumber:=" , "",
      "Version:=" , "1.0",
      "Notes:=" , "",
```


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

```
      "OwnerName:=" , "username",
      "OwnerEmail:=" , "user.name@company.com",
      "Date:=" , "9:12:32 AM Oct 27, 2025"
    ],
    [
      "NAME:ComponentSecurityData",
      "IsEditAllowed:=" , True,
      "NewEditPassword:=" , "111111"
    ]
  ],
  [
    "NAME:CreateRepresentationsData",
    [
      "NAME:RepresentationData",
      [
        "NAME:HeaderAndGeometryData",
        "SourceDesignName:=" , "Waveguide_Filter:IcepakDesign1",
        "RepresentationName:=" , "IcepakDesign1",
        "IsEncrypted:=" , True,
        "PasswordType:=" , "InternalPassword",
        "VisibleParts:=" , ["wg_flange_in"],
      ]
    ],
    [
      "NAME:RepresentationData",
      [
        "NAME:HeaderAndGeometryData",
        "SourceDesignName:=" , "Waveguide_Filter:filter",
```


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

```
        "RepresentationName:=" , "filter",
      ]
    ]
  ],
  "D:/Designs/ComponentCmd/test2.acmp2")
```

EditComponent2_0

After creating a 2.0 component, you can use the EditComponent2_0 command to add or delete design representations, change the component's information, or change its security.

> **Important:** A representation cannot be edited once it is committed to a component.

Note that this topic includes two examples of how to use the Edit Component2_0 command: the first example is a typical script that would be generated from the **Tools** > **Record Script to File** user interface function. The second example is a simplified version of the EditComponent2_0 command that includes only the required script functions needed to create the component; this version can aid in automation.

> **Note:**
>
> A recorded script does not record Component 2.0 passwords; they must be added after the script is created.

UI Access	Right click on a design in the Project Manager tree and select **Edit Component 2.0**, then select the Component 2.0 file to edit.
Parameters	

Name	Type	Description
<Filename>	String	Filepath of the Component 2.0 file to be edited; it has the .acmp2 file extension.
<ComponentData>	Array	Structured array that defines the properties of the top-level 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.

			component. It has the following parameters:
			`["NAME:ComponentData",`
			`[ComponentHeaderInfoData],` # Array that sets the information for the component file; see example below.
			`[ComponentAuthenticationData]` # Needed if the 2.0 component is encrypted; see example below.
			`[ComponentSecurityData],` # Needed if you want to change the security settings on the component file; it is an array that defines the overall security for the 2.0 component; see example below.
			`]`

	<CreateRepresentations Data>	Array	Needed if you want to add a representation to the specified 2.0 component. It is a structured array that defines the representation in the 2.0 component. There can be multiple <CreateRepresentationsData> definitions. It has the following parameters:
			`["NAME:CreateRepresentationsData",`
			`["NAME:RepresentationData",`
			`[HeaderAndGeometryData]` # Array that defines the representation parameters defined on the **Info**, **Model**, **Coordinate Systems**, **Parameters**, and **Encryption** tabs of the Create Component 2.0 Representation window; see the example below.
			`[DesignData]` # Array that defines parameters on the design-specific tabs; which depending on the design may be **Boundaries**, **Excitations**, **Circuit Elements**, **Hybrid Regions**, and **Mesh** tabs of the Create Component 2.0 Representation window; see the example below.
			`[ImageFile]` # Optional array that defines the image to display in the upper right area of the 3D modeler whenever the representation is used. The filename is a string that includes the filepath 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.

			]]
	<DeleteRe-presentationsData>	Array	Needed if you want to delete a representation from the specified 2.0 component. You must specify the representation name (as a string), and if the representation is password protected, you must specify the password. See the example below.
Return Value	None		

Python Syntax	EditComponent2_0(<*ComponentData*>, <*CreateRepresentationsData*>, <*Filename*>)

Python Example	The example below demonstrates the creation of the 2.0 component and the addition of two representations.
	`oDesktop.OpenProject("D:/Designs/Waveguide_Filter.aedt")`
	`oDesktop.EditComponent2_0(`
	`"D:/Design/AAProjects/ComponentCmd/test.acmp2",`
	`[`
	`"NAME:ComponentData",`
	`[`
	`"NAME:ComponentHeaderInfoData",`
	`"ComponentName:=" , "Filter1",`
	`"Company:=" , "",`
	`"CompanyURL:=" , "",`
	`"ModelNumber:=" , "",`
	`"Version:=" , "1.01",`
	`"Notes:=" , "This is test note",`

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

```
      "OwnerName:=" , "username",
      "OwnerEmail:=" , "user.name@company.com",
      "Date:=" , "9:12:32 AM Nov 18, 2025"
    ],
    [
      "NAME:ComponentAuthenticationData", # needed if the component is pass-
      word protected
      "ComponentEditPassword:=", "111111"
    ],
    [
      "NAME:ComponentSecurityData",# changing the password
      "IsEditAllowed:=" , True,
      "NewEditPassword:=" , "222222"
    ]
  ],
  [
    "NAME:CreateRepresentationsData", # adding a representation
    [
      "NAME:RepresentationData",
      [
        "NAME:HeaderAndGeometryData",
        "SourceDesignName:=" , "Waveguide_Filter:filter",
        "RepresentationName:=" , "filter22",
        "Company:=" , "",
        "Company URL:=" , "",
        "Model Number:=" , "",
        "Help URL:=" , "",
        "Version:=" , "1.0",
        "Notes:=" , "",
        "IconType:=" , "",
```


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

```
"Owner:=" , "username",
"Email:=" , "user.name@company.com",
"Date:=" , "4:13:02 PM Nov 18, 2025",
"HasLabel:=" , False,
"IsEncrypted:=" , False,
"AllowEdit:=" , False,
"SecurityMessage:=" , "",
"Password:=" , "",
"EditPassword:=" , "",
"PasswordType:=" , "UnknownPassword",
"HideContents:=" , True,
"ReplaceNames:=" , True,
"ComponentOutline:=" , "None",
"IncludedParts:=" , ["wg_flange_in","wg_inner","wg_outer","wg_
flange_out2224455"],
"VisibleParts:=" , [],
"IncludedCS:=" , [],
"DefaultHandle:=" , "Global",
"IncludedParameters:=" , ["fillet_radius","fillet_radius_
flange","len_wg1","len_wg2","len_wg3","t1","t2","t_
flange","w1","w2","w_flange","wg_height","wg_thickness","wg_width"],
"IncludedDependentParameters:=", [],
"ParameterDescription:=", [ "fillet_radius:=" , "", "fillet_radius_
flange:=", "", "len_wg1:=" , "", "len_wg2:=" , "", "len_wg3:=" , "",
 "t1:=" , "", "t2:=" , "", "t_flange:=" , "", "w1:=" , "", "w2:=" ,
"", "w_flange:=" , "", "wg_height:=" , "", "wg_thickness:=" , "",
"wg_width:=" , ""],
"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:=" , ["1","2"]
    ],
    [
      "NAME:ImageFile",
      "ImageFile:=" , ""
    ]
  ],
  [
    "NAME:RepresentationData",
    [
      "NAME:HeaderAndGeometryData",
      "SourceDesignName:=" , "Waveguide_Filter:IcepakDesign1",
      "RepresentationName:=" , "IcepakDesign1",
      "Company:=" , "",
      "Company URL:=" , "",
      "Model Number:=" , "",
      "Help URL:=" , "",
      "Version:=" , "1.0",
      "Notes:=" , "",
      "IconType:=" , "",
      "Owner:=" , "username",
      "Email:=" , "user.name@company.com",
      "Date:=" , "4:12:58 PM Nov 18, 2025",
      "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:=" , ["wg_flange_in","wg_inner","wg_outer","wg_
        flange_out"],
        "VisibleParts:=" , [],
        "IncludedCS:=" , [],
        "DefaultHandle:=" , "Global",
        "IncludedParameters:=" , [],
        "IncludedDependentParameters:=", [],
        "ParameterDescription:=", [],
        "IsLicensed:=" , False,
        "LicensingDllName:=" , "",
        "VendorComponentIdentifier:=", "",
        "PublicKeyFile:=" , ""
      ],
      [
        "NAME:DesignData"
      ],
      [
        "NAME:ImageFile",
        "ImageFile:=" , ""
      ]
```


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

	``
	    ]
	  ],
	  [
	    "NAME:DeleteRepresentationsData", #deleting a representation
	    [
	      "NAME:DeleteRepresentationData",
	      "RepresentationName:=" , "filter"
	    ],
	    [
	      "NAME:DeleteRepresentationData",
	      "RepresentationName:=" , "IcepakDesign1",
	      "RepresentationEditPassword:=", "111111" #needed if representation is
	      password protected
	    ]
	  ])
	``

Python Example	This example is a simplified version of the EditComponent2_0 command; it includes only those functions that are necessary to edit the component, but users can add other functions (as shown above) as desired.
	``
	oDesktop.EditComponent2_0(""D:/Designs/test2.acmp2",
	  [
	    "NAME:ComponentData",
	    [
	      "NAME:ComponentAuthenticationData",
	      "ComponentEditPassword:=", "111111"
	    ]
	  ],
	  [
	    "NAME:CreateRepresentationsData",
	    [
	``

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

```
        "NAME:RepresentationData",
        [
          "NAME:HeaderAndGeometryData",
          "SourceDesignName:=" , "Waveguide_Filter:IcepakDesign1",
          "RepresentationName:=" , "IcepakDesign2"
        ]
      ]
    ],
    [
      "NAME:DeleteRepresentationsData"
    ])
```

InsertComponent2_0

This command is used to insert a representation from a 2.0 component.

> **Note:**
>
> This command is not an oDesktop command, but it has been included here because it is tightly associated with the CreateComponent2_0 command, which is an oDesktop command.

UI Access	Right click on a design in the Project Manager tree and select **Insert Component 2.0**
Parameters	

Name	Type	Description
<FileName>	String	Specifies the filepath of the 2.0 component that has the desired representation
<RepresentationName>	String	Name of the representation to insert

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

	<RepresentationId>	Integer	With this release, the only value accepted is -1.
Return Value	Boolean: True if placed; False if not placed.		

Python Syntax	InsertComponent2_0(*<FileName>*, *<RepresentationName>*,*<RepresentationId>*)

Python Example	`oDesign = oProject.SetActiveDesign("HFSSDesign1")`
	`oDesign.InsertComponent2_0(`
	`  "D:/Designs/insCmdsAllDesigns.acmp2",`
	`  "Filter1",`
	`  -1)`

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.

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

InsertDesignWithWorkflow

InsertToolObject

Paste [Project Object]

Redo [Project Level]

RemoveAllUnusedDefinitions

RemoveMaterial

RemoveUnusedDefinitions

Rename

RunToolkit

Save

SaveAs

SaveAsStandAloneProject

SaveProjectArchive

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

SetActiveDefinitionEditor

SetActiveDesign

SetPropValue [Project]

SetPropertyValue

SetVariableValue

SimulateAll

Undo [Project]

UpdateDefinitions

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>

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.

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.

Python Syntax	ClearMessages()
Python Example	`oProject.ClearMessages()`

Close

Closes the active project.

> **Warning:**
>
> Unsaved changes will be lost.

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	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.

Python Syntax	CopyDesign (*<DesignName>*)
Python Example	`oProject.CopyDesign ("HFSSDesign1")`

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

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")`

DeleteDataset

Deletes a specified dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Remove**.

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 <*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.

Python Syntax	DeleteDesign (<*DesignName*>)
Python Example	`oProject.DeleteDesign("IcepakDesign2")`

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

DeleteToolObject

> **Note:**
> This command is for internal Ansys use only.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ObjectName*></td><td>String</td><td>Name of the tool object.</td></tr></table>
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.

UI Access	**Project** > **Datasets** > **Edit**.
Parameters	<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>

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	EditDataset (<*OriginalName*> <*DatasetDataArray*>)

Python Example	oProject.EditDataset ("ds1"
	["NAME:ds2",
	  ["NAME:Coordinates",
	  [
	  "NAME:Coordinate",
	  "X:=", 1, "Y:=", 2
	  ],
	  [
	  "NAME:Coordinate",
	  "X:=", 3, "Y:=", 4
	  ]
	  ]
	]
	)
	oDesign.EditDataset ("ds1"
	["NAME:ds2",
	  ["NAME:Coordinates",
	  [

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

```
        "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
<*OriginalName*>	String	Name of the material before editing.
<*MatProperties*>	Array	Structured array containing material properties:

```
[
  "NAME:<New material name>",
  "CoordinateSystemType:=", <string>,
```


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

		"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>]
<ModifierDataArray>	Array	Optional structured array containing thermal or spatial modifiers: ["NAME:ModifierData", ["NAME:<ThermalModifierData or SpatialModifierData>", "modifier_data:=" , <"thermal_modifier_data" or "spa-

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

```
    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>,
      ]
    ]
  ]
]
```

Return Value	None.

Python Syntax	EditMaterial (<*OriginalName*>, <*MatProperties*>)

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

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",
  "thermal_expansion_coeffcient:=", "7.2e-006"
]
)
```

With Thermal Modifier:

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

```
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",
        [
        "NAME:one_thermal_modifier",
        "Property::=" , "permittivity",
        "Index::=" , 0,
```


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",
          "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

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Drude_plasma_parameters_r231")
```


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()
oDefinitionManager.EditMaterial("Drude",
  [
    "NAME:Drude",
    "CoordinateSystemType:=", "Cartesian",
    "BulkOrSurfaceType:="   , 1,
    [
      "NAME:PhysicsTypes",
      "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",
```


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

```
            "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')` `oDesign.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.

ExportMaterial

Exports a local material to a library.

UI Access	**Export to Library** command in the material editor.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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.

> **Note:** `GetActiveDesign` 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	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>* — Type: String — Description: (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.

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><*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()
Python Example	`oProject.GetChildTypes()`

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

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
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	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: *<designName>*, Type: String, Description: 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()`

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*includeReadOnly*></td><td>Boolean</td><td>(Optional) • True – Include read only props. • False – Do not include read only props.</td></tr></table>
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.

GetPropValue [Project]

Returns the property value for the active project object, or specified property values.

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: Property Function Summary.</td></tr></table>
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	<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") </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.

			• 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.

> **Tip:** Use the script recording feature and edit a property, and then view the resulting script to see the format for that property.

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><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>String value of the property.</td></tr>
</table>

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)
Python Example	`selectionArray = oEditor.GetSelections()` `for k in selectionArray:`

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

	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")` `...`

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.

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 represents the value of the variable.

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.

HasDataset

Determines whether a specified dataset exists.

UI Access	N/A
Parameters	Name: *<datasetName>* — Type: String — Description: Name of the specified dataset.
Return Value	Integer: • **1** - dataset exists. • **0** - dataset does not exist.

Python Syntax	HasDataset (*<datasetName>*)
Python Example	`oDesign.HasDataset('$ds1')`

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	Name: *<datasetFileFullPath>* — Type: String — Description: The full path to the file containing the dataset values. *.tab files recommended (see note below).

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

	<*optionalDatasetName*>	String	*Optional*. User-defined dataset name.
Return Value	None.		

Python Syntax	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)

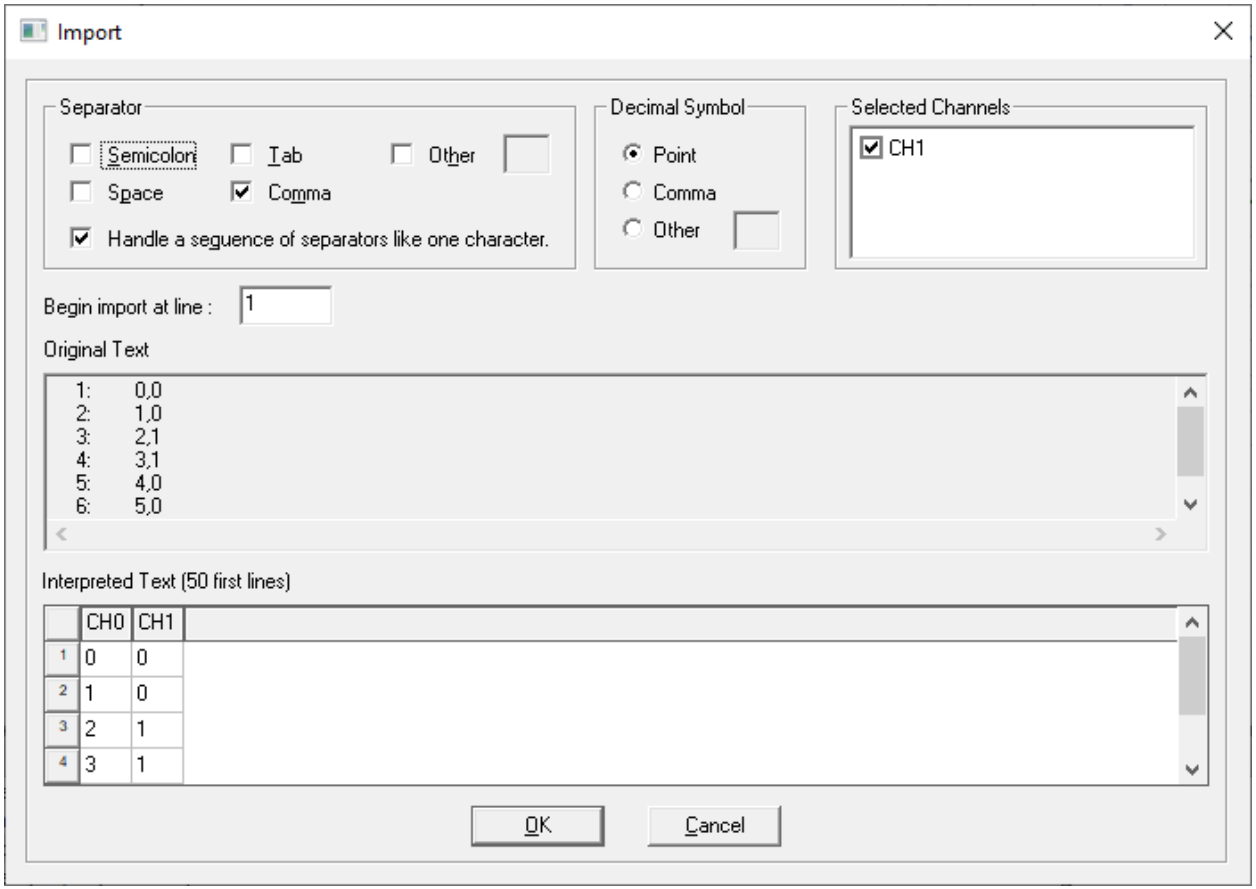
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

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.



InsertDesign

Inserts a new design in the project. For Icepak 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 Icepak Design]**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignType*></td><td>String</td><td>Design type."Icepak".</td></tr><tr><td><*DesignName*></td><td>String</td><td>Design name.</td></tr><tr><td><SolutionType><ProblemType></td><td>String</td><td>Can be "SteadyState TemperatureAndFlow", "SteadyState TemperatureOnly", "SteadyState FlowOnly", "Transient TemperatureAndFlow", "Transient TemperatureOnly", or "Transient FlowOnly" .</td></tr><tr><td>""</td><td>None</td><td>Empty argument.</td></tr></table>
Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*SolutionType*>,'')
Python Example	`oProject.InsertDesign('Icepak','MyDesign', 'TemperatureAndFlow', '')`

InsertDesignWithWorkflow

Inserts a design with a named workflow and returns an IDispatch string.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*type*></td><td>String</td><td>Type of design.</td></tr><tr><td><*workflowName*></td><td>String</td><td>Name of the workflow.</td></tr><tr><td><*specName*></td><td>String</td><td>Name of the spec.</td></tr><tr><td><*fileName*></td><td>String</td><td>Name of the file.</td></tr><tr><td><*libLoc*></td><td>String</td><td>Type of library, such as SysLib.</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.

	<stationaryPath>	String	Path.
Return Value	IDispatch string, such as 'IDispatch(IAltraSimScript)'		

Python Syntax	InsertDesignWithWorkflow(<*type*>, <*workflowName*>, <*specName*>, <*fileName*>, <*libLoc*>, <*stationaryPath*>)
Python Example	`oProject.InsertDesignWithWorkflow ("Circuit Design", "Serial Design", "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.

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** > **Paste**.
Parameters	None.
Return Value	None

Python Syntax	Paste()
Python Example	`oProject.Paste()`

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.

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	None.

Python Syntax	RemoveAllUnusedDefinitions()
Python Example	`oProject.RemoveAllUnusedDefinitions()`

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*>)

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.RemoveMaterial ([` `"Material1", false, "mo0907","UserLib"])`

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",`

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

	"Steel-oxidised-surface"]])

Rename

Renames the project and saves it. Similar to SaveAs().

UI Access	**Edit** > **Rename**.
Parameters	Name / Type / Description <*NewName*> / String / Desired name of the project. The path is optional. <*OverWriteOk*> / Boolean / • True - overwrite the file on disk if it exists. • False - prevent overwrite.
Return Value	None.

Python Syntax	Rename(<*NewName*>,<*OverWriteOK*>)
Python Example	`oProject.Rename("c:\projects\MyProject.aedt", True)`

Save

Saves 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.

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.

> **Note:**
>
> This script takes two parameters for non-schematic/layout designs and four parameters for schematic/layout designs.

UI Access	**File** > **Save As**.
Parameters	Name / Type / Description *<NewName>* / String / The desired name of the project, with directory and extension. *<OverWriteOK>* / Boolean / True to overwrite the file of the same name, if it exists. False to prevent overwrite. *<DefaultAction>* / String / For Schematic/Layout projects only. Otherwise omit. See note below. Valid actions: ef_overwrite , ef_copy_no_overwrite, ef_make_path_abso-

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

			lute, or empty string.
	<*OverwriteActions*>	Array	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.
Return Value	None.		

Python Syntax	For non-Schematic/Layout project: SaveAs (<*NewName*> <*OverWriteOK*>) For Schematic/Layout project: SaveAs (<*NewName*> <*OverWriteOK*> <*DefaultAction*> <*OverrideActions*>)

Python Example	`oProject.SaveAs('D:/projects/project1.aedt', True)`
	---- 	
	`oProject.SaveAs('D:/Projects/Project1.aedt', True, 'ef_overwrite', ['NAME:OverrideActions', ['NAME:ef_copy_no_overwrite', ['NAME:Files',`
	`'$PROJECTDIR/circuit_models.inc']], ['NAME:ef_make_path_absolute', ['NAME:Files', '$PROJECTDIR/SL_6s.sp']]])`

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

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.

<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><projectName></td><td>String</td><td>The desired name of the project, with directory and extension.</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	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*>)
Python Example	`oProject.SaveProjectArchive("C:\\Users\\Documents\\Ansoft\\Project27.aedtz", True, False, [], "")`

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

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.

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.

	<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")` `oProject.SetPropValue("TeeModel/Results/S Parameter Plot 1/Display Type", "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.

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>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 Compłnst@R;2;3</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.

	<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("IcepakDesign1")
```

Conventions Used in this Chapter

`<ModuleName>` is a placeholder for any one of the following modules:

- Analysis Module – "AnalysisSetup"
- Boundary Module – "BoundarySetup"
- Field Overlays Module – "FieldsReporter"
- Mesh Module – "MeshSetup"
- Optimetrics Module – "Optimetrics"
- Reporter Module – "ReportSetup"
- Solutions Module – "Solutions"

AddDataset

AddDynamicLink

AddModelingProperties

AnalyzeAll

ApplyMeshOps

ChangeProperty

ConstructVariationString

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

CopyArray

CopyItemCommand

CreatePCBPowermap

CreateReport

DeleteDataset

DeleteFieldVariation

DeleteFullVariation

DeleteLinkedDataVariation

DeleteOutputVariable

DeletePlotEntities

DeleteSolutionVariation

EcxmlExport

EditDesignSettings

EditNotes

ExportConvergence

ExportLayoutViaCurrentDensity

ExportLPVROM

ExportNetworkCSV

ExportProfile

ExportReport

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

FitSelected

GenerateConductivityFile

GenerateMesh

GetAllPorts

GetChildNames [Design]

GetChildObject [Design]

GetChildTypes [Design]

GetDesignType

GetManagedFilesPath

GetModule

GetName

GetNominalVariation

GetNoteText

GetProject

GetPostProcessingVariables

GetPropNames [Design]

GetPropValue [Design]

GetSelections

HideGeometry

ImportNetworkCSV

PasteDesign

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

Redo

RenameDesignInstance

RenameSource

RevertAllToInitialCondition

RunToolkit

RunCTIMAPI

SelectObjects

SetActiveEditor

SetDesignSettings

SetPropValue [Design]

SetPropertyValue

SetShowLayoutForLayoutComponent

ShowGeometry

Undo

ValidateDesign

AddDataset

Adds a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Add**		
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.

	<*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:$ds1",`
	`  ["NAME:Coordinates",`
	`  ["NAME:Coordinate",`
	`  "X:=", 2,`
	`  "Y:=", 4`
	`  ],`
	`  ["NAME:Coordinate",`
	`  "X:=", 6,`
	`  "Y:=", 8`
	`  ]`
	`  ]`
	`]`
	`)`
	`oDesign.AddDataset(`
	`[`
	`"NAME:$ds1",`

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

```
   ["NAME:Coordinates",
     ["NAME:Coordinate",
       "X:=", 2,
       "Y:=", 4
     ],
     ["NAME:Coordinate",
       "X:=", 6,
       "Y:=", 8
     ]
   ]
]
)
```

AddModelingProperties

Use: Add a modeling property to a design

Command: None

Syntax: AddModelingProperties <design>

Return Value: None

Parameters: <design>

Type: string

AnalyzeAll [design]

Runs all solution setups and Optimetrics setups for the current design instance.

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>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	AnalyzeAll()
Python Example	`oDesign.AnalyzeAll(isBlocking)`

AnalyzeAllNominal

Runs all defined setups.

UI Access	Right-click **Analysis** in the project tree, and select **Analyze All**.
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>
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	AnalyzeAllNominal()
Python Example	`oDesign.AnalyzeAllNominal()`

ApplyMeshOps

This script has been replaced by GenerateMesh.

ConstructVariationString

Lists and orders the variables and values associated with a design variation.

UI Access	N/A
Parameters	Name / Type / Description <*ArrayOfVariableNames*> / Array of Strings / List of variable names. <*ArrayOfVariableValuesIncludingUnits*> / Array of Strings / List of variable values, including units, in the same order as the list of names.
Return Value	Returns variation string with the variables ordered to correspond to the order of variables in design variations. The values for the variables are inserted into the variation string. For an example of how ConstructionVariationString can be used, see the Python example.

Python Syntax	ConstructVariationString(<*ArrayOfVariableNames*>, <*ArrayOfVariableValuesIncludingUnits*>)
Python Example	`varStr = oDesign.ConstructVariationString(["xx", "yy"], ["2mm", "1mm"])` `oDesign.ExportProfile("Setup1",  varStr, "C:\profile.prof")`

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

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

CreatePcbPowermap

Creates sources based on EM loss data (powermap) linked to a PCB native component from a source HFSS 3D layout design.

<table>
<tr><td>UI Access</td><td colspan="3">Icepak > Visualization > PCB Map to Source</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><NAME></td><td>string</td><td>NA</td></tr>
<tr><td><PCB Name></td><td>string</td><td>Name of the selected PCB native component.</td></tr>
<tr><td><Number of Rows></td><td>int</td><td>Number of rows to divide the source powermap into.</td></tr>
<tr><td><Number of Columns></td><td>int</td><td>Number of columns to divide the source powermap into.</td></tr>
<tr><td><Minimum Power></td><td>int</td><td>Minimum power to filter out of the sources created.</td></tr>
<tr><td><Use Existing Solution></td><td>bool</td><td>True or False</td></tr>
<tr><td><Export Directory></td><td>int</td><td>If Use Existing solution is False, enter a a file path to the powermap file.</td></tr>
<tr><td><Source Position></td><td>string</td><td>Bottom, Middle, or Top.</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	CreatePcbPowermap (<*NAME*>, <*PCB Name*>, <*Number of Rows*>, <*Number of Columns*>, <*Minimum Power*>, <*Use Existing Solution*>, <*Export Directory*>, <*Source Position*>)

Python Example	`oDesign.CreatePcbPowermap(`
	`  [`
	`  "NAME:PCBPowerMapData",`
	`  "PCB Name:=" , "PCB1_1",`
	`  "Number of Rows:=" , 10,`
	`  "Number of Columns:=" , 20,`
	`  "Minimum Power:=" , 1E-05,`
	`  "Use Existing Solution:=", True,`
	`  "Export Directory:=" , "C:/Users/spsander/OneDrive - ANSYS, Inc/Desktop/AEDT/R26.1/S1266083/Project Files/PCB/2_vias_dc.aedtresults/IcepakDesign2.results/CV239_LINK111_V0/",`
	`  "Source Position:=" , "Middle"`
	`  ])`

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 with ports.

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

			"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 Stacked Plot" or "3D Polar Plot"`
			
			If <ReportType> is "Emission Test", then set to 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.

			"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",`
	`[`
	`"NAME:Context",`

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

```
                "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))"
                     ],
```


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: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.

```
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
				]
			]
		]
	])

oModule.ChangeProperty(
	[
		"NAME:AllTabs",
		[
			"NAME:Bar",
			[
				"NAME:PropServers",
				"S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
```


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")
```

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.

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	DeleteDataset (<*DatasetName*>)
Python Example	`oProject.DeleteDataset('$ds1')` `oDesign.DeleteDataset('$ds1')`

DeleteOutputVariable

Deletes an existing output variable. The variable can only be deleted if it is not in use by any traces.

UI Access	**Icepak** > **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.

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**, 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"],`
	`  ["width='2in'", "Setup1", "Sweep1"]`
	`])`

DeleteOutputVariable

Deletes an existing output variable. The variable can only be deleted if it is not in use by any traces.

UI Access	**Icepak** > **Results** > **Output Variables**. In the **Output Variables** window, click **Delete**.
Parameters	<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></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	DeleteOutputVariable (<*OutputVarName*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.DeleteOutputVariable ("testNew")`

EditCoSimulationOptions

Sets options for cosimulation.

Command: None

Syntax: EditCoSimulationOptions <array_name>

Return Value: None

Parameters: <array_name>

Type: string

EditDesignSettings

Edits design settings.

UI Access	[**Icepak**] > **Design Settings**.		
Parameters	Name	Type	Description
	<*overrideArray*>	Array	Structured array. The available settings are determined by the active solver.
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	EditDesignSettings(<*overrideArray*>)

Python Example	`oDesign.EditDesignSettings(`
	
	`  [`
	`  "NAME:Design Settings Data",`
	`  "Allow Material Override:=", True,`
	`  "Perform Minimal validation:=", False,`
	`  "EnabledObjects:=" , [],`
	`  ],`
	`  [`
	`  "NAME:Model Validation Settings",`
	`  "EntityCheckLevel:=" , "Basic",`
	`  "IgnoreUnclassifiedObjects:=", False,`
	`  "SkipIntersectionChecks:=", False,`
	`  "SkipMeshChecks:=", True`
	`  ])`

EditInfiniteArray

Edits the properties of an infinite array

Command: None

Syntax: EditInfiniteArray <array_name>

Return Value: None

Parameters: <array_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.

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>,
"LowPriorityForSimulations:=", <boolean>,
"SaveNearFieldSolutions:=", <boolean>,
"SchematicEnabled:=", <boolean>,
"UseGlobalNumProc:=", <boolean>,
"ComputeBothEvenAndOddCPWModes:=", <boolean>,
"NumProcessors:=", <int>,
"NumProcessorsDistrib:=", <int>,
"CausalMaterials:=", <boolean>,
"UseHPCForMP:=", <boolean>,
"HPCLicenseType:=", <int>)

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

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

HPCLicenseType - number indicating hpc license type

ExportConvergence

For a given variation, exports convergence data (max mag delta S, E, freq) to *.conv file.

<table>
<tr><td>UI Access</td><td colspan="3">Results > Solution Data. Select Convergence tab and click Export.</td></tr>
<tr><td>Parameters</td><td colspan="3"><table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><<i>Setup</i>></td><td>String</td><td>Setup name.</td></tr>
<tr><td><<i>VariationKeys</i>></td><td>String</td><td>The variation. Pass empty string for the current nominal variation.</td></tr>
<tr><td><<i>FilePath</i>></td><td>String</td><td>Full path to desired *.conv file location.</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	
Python Example	

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')`

ExportLPVROM

Exports LPV ROM files for use in TwinBuilder.

UI Access	**Icepak** > **Export ROM** > **LPV ROM**

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><*Parametric Setup*></td><td>string</td><td>Parametric Setup</td></tr><tr><td><*Transient Setup*></td><td>string</td><td>Transient Setup</td></tr><tr><td><*Flie path*></td><td>string</td><td>The file path to the AEDT project export location</td></tr></table>
Return Value	None

Python Syntax	ExportLPVROM (<*C*>)
Python Example	`oDesign.ExportLPVROM("ParametricSetup", "TransientSetup", "C:/Model/LPVProject_LPVROM.aedtexport/")`

EcxmlExport

Exports and Electronics Cooling (EC XML) file.

UI Access	**Icepak** > **Export ECXML**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>NA</td><td>NA</td><td>NA</td></tr></table>
Return Value	None

Python Syntax	EcxmlExport
Python Example	`oDesign.EcxmlExport()`

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

ExportNetworkCSV

Exports a selected network as a comma-separated value (.csv) file.

UI Access	**Icepak** > **Export** > **Network CSV**
Parameters	Name: *<C>* — Type: int — Description: File path to the location to save the network .csv file.
Return Value	None

Python Syntax	ExportNetworkCSV (*<C>*)
Python Example	`oDesign.ExportNetworkCSV("C:\\Desktop\\AEDT\\network_data")`

ExportProfile

Exports a solution profile to file.

UI Access	N/A
Parameters	Name: *<SetupName>* — Type: String — Description: Text of the design's notes. Name: *<VariationString>* — Type: String — Description: Variation name. Pass empty string for the current nominal variation. Name: *<filePath>* — Type: String — Description: Full file path, including extension *.prof.
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	
Python Example	

FitSelected

Zooms to fit the specified object in the 3D Modeler window.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	FitSelected()
Python Example	`oDesign.FitSelected("[34]")`

GenerateConductivityFile

Creates a conductivity file for a specified PCB or layout component.

UI Access	None
Parameters	Name | Type | Description <*GenerateConductivityFile*> | string | PCB or layout component 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	GenerateConductivityFile ()
Python Example	`oDesign.GenerateConductivityFile("PCB1_1")`

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	[**Icepak**] > **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'])`

GetChildNames [Design]

Returns the names of the design's child objects.

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><*type*></td><td>String</td><td>Optional. Valid options are "Module", "Editor", and "Variable". Default returns Module names.</td></tr></table>
Return Value	Names of children for the queried object).

#160;

Python Syntax	GetChildNames (<*type*>)
Python Example	`oDesign.GetChildNames("Variable")`

GetChildObject [Design]

Returns the design's child objects.

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	A child object if one is found. Otherwise script error.

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")`

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

	`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.

UI Access	Not Available
Parameters	None
Return Value	String indicating 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.

Python Syntax	GetDesignID()
Python Example	`oDesign = oProject.GetActiveDesign()` `oDesign.GetDesignID()`

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()`

GetEdgePositionAtNormalizedParameter

Gets the position on an edge with normalized parameter.

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.

	<EdgeID>	Integer	Edge ID.
	<NormParam>	Double	Normalized parameter.(Proportional to the length of the specified edge) For example, 0 leads to the start vertex position of the edge, 1 leads to the end vertex position of the edge, 0.5 leads to the mid position on the edge.
Return Value	Array of string containing the x, y and z coordinate values.		

Python Syntax	GetEdgePositionAtNormalizedParameter(*<EdgeID>*, *<NormParam>*)

Python Example	`oEditor.GetEdgePositionAtNormalizedParameter(5, 0)`
	`oEditor.GetEdgePositionAtNormalizedParameter(5, 1)`
	`oEditor.GetEdgePositionAtNormalizedParameter(5, 0.5)`

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.

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

	<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")`
	`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

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

	<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")`
	`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.

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 path where project results are located.

Python Syntax	oDesign.GetManagedFilesPath()
Python Example	`oDesign.GetManagedFilesPath()`

GetModule

Returns the IDispatch for the specified module.

UI Access	N/A
Parameters	Name: *<modulename>* Type: String Description: One of the following: • Analysis Module – "AnalysisSetup" • Boundary Module – "BoundarySetup" • Field Overlays Module – "FieldsReporter" • Mesh Module – "MeshSetup" • Optimetrics Module – "Optimetrics" • Reporter Module – "ReportSetup" • Solutions Module – "Solutions"
Return Value	Module 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	GetModule (<*modulename*>)
Python Example	`oModule = oDesign.GetModule("SimSetup")`

GetName

Returns the design name of the active design, in that order separated by a semicolon.

UI Access	N/A
Parameters	None.
Return Value	String indicating the name of the active design.

Python Syntax	GetName()
Python Example	design_name = oDesign.GetName()

GetNominalVariation

Returns the current nominal variation.

UI Access	N/A
Parameters	None.
Return Value	String containing 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.

Python Syntax	GetNominalVariation()
Python Example	`oDesign.GetNominalVariation()`

GetNoteText

Returns the text of the note attached to a design.

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.

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	GetObjPath()
Python Example	`oDesign.GetObjPath()`

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.

UI Access	N/A
Parameters	See table below
Return Value	Double value of the output variable.

Name	Type	Description
<*OutputVarName*>	String	Name of the output variable.
<*IntrinsicVariation*>	String	A set of intrinsic variable value pairs to use when evaluating the output expression.
<*SolutionName*>	String	Name of the solution as listed in the output variable UI. For example, `"Setup1 : Last Adaptive"`.
<*ReportType*>	String	The name of the report type as seen in the output variable UI.
<*ContextArray*>	Array	Structured array containing context for which the output variable expression is being evaluated. Can be empty. `["Context:=", <string>]`

Python Syntax	GetOutputVariableValue(<*OutputVarName*>, <*IntrinsicVariation*>, <*SolutionName*>, <*ReportTypeName*>,

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

	<*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()`

GetPostProcessingVariables

Returns the list of post-processing variables.

UI Access	N/A
Parameters	None.
Return Value	Returns array containing variables.

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	GetPostProcessingVariables()
Python Example	`oDesign.GetPostProcessingVariables()`

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	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 CompInst@R;2;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	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

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")

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

			• 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 [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	Type	Description
	<includeReadOnly>	Boolean	(Optional). • True - include read only property.

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

			• 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.

UI Access	N/A		
Parameters	Name	Type	Description
	<*propPath*>	String	A child object's property path. See: Object Property Script Function Summary.
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')`

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

GetSelections [Design]

This script serves no function at the Design level. See: GetSelections (Layout Editor), GetSelections (Model Editor), or Get Selections (Schematic Editor).

GetSolutionType

Returns solution type of the design.

UI Access	N/A
Parameters	None.
Return Value	String containing the solution type. Possible values are

Python Syntax	GetSolutionType()
Python Example	`oDesign.GetSolutionType()`

GetSolveInsideThreshold

Returns the solve inside threshold. This command does not apply to HFSS-IE.

UI Access	N/A
Parameters	None.
Return Value	Double representing the solve inside threshold.

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	GetSolveInsideThreshold()
Python Example	`oDesign.GetSolveInsideThreshold()`

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()`

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.

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 represents the value of the variable.

Python Syntax	GetVariableValue(<*VarName*>)
Python Example	`oProject.GetVariableValue("var_name")`

GetVariationVariableValue

Returns the value for a specified variation's variable.

UI Access	N/A
Parameters	Name / Type / Description <*VariationString*> / String / The name of the design variation. <*VariableName*> / String / The name of the variable.
Return Value	Returns a double precision value in SI units, interpreted to mean the value of the variable contained in the variation string.

Python Syntax	GetVariationVariableValue(<*VariationString*>, <*VariableName*>)
Python Example	`oDesign.GetVariationVariableValue('x_size = 2mm y_size = 1mm', 'y_size')`

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

HideGeometry

Disables visualization for the specified object.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	oDesign.HideGeometry()
Python Example	`oDesign.HideGeometry("[34]")`

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	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.

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	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)

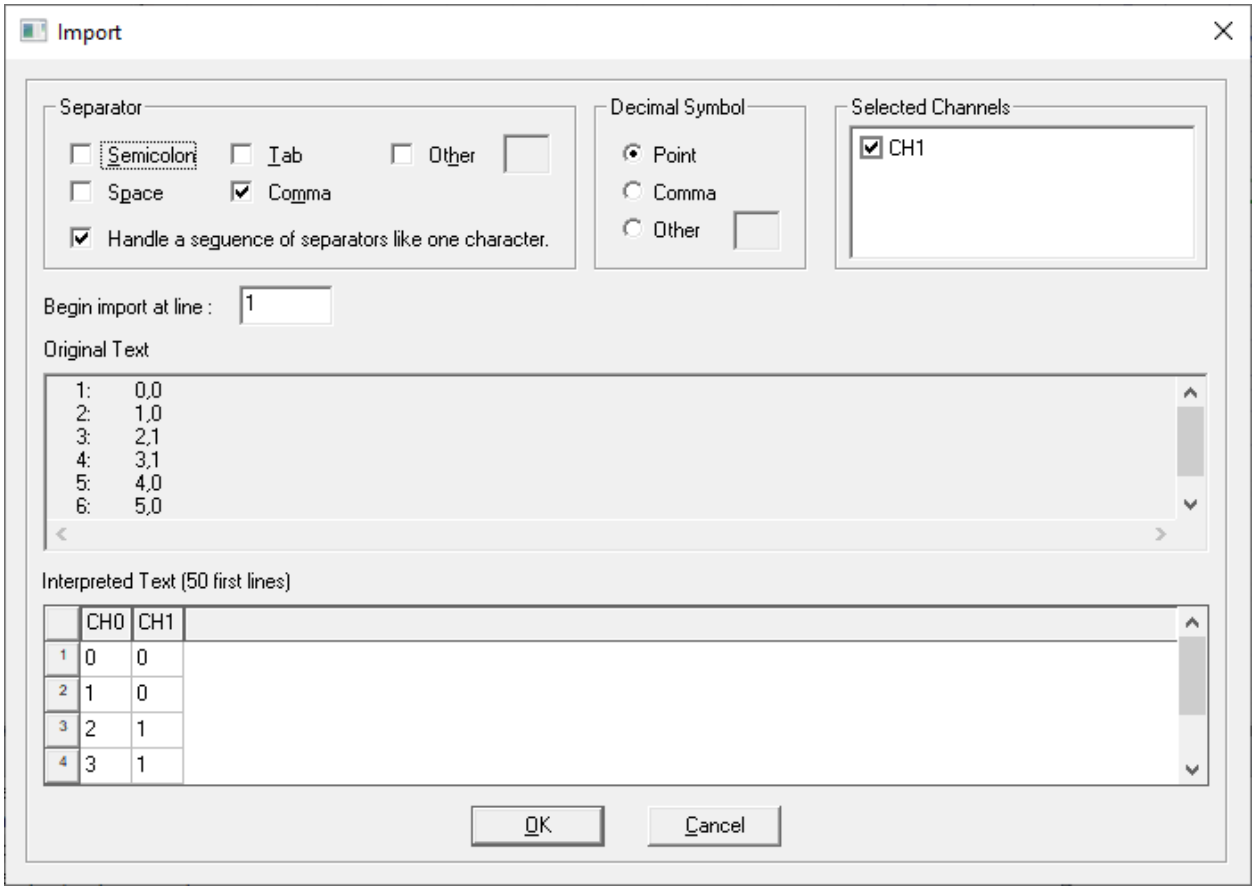
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

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.



ImportNetworkCSV

Imports a selected network from a comma-separated value (.csv) 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	**Icepak** > **Import** > **Network CSV**
Parameters	Name | Type | Description <*C*> | int | File path to the location of the network .csv file.
Return Value	None

Python Syntax	ImportNetworkCSV (<*C*>)
Python Example	`oDesign.ImportNetworkCSV("C:\\Project Files\\Networks\\network_data_curve")`

InitializeSystemCoupling

Used for the System Coupling workflow. It generates a System Coupling configuration (.spc) file and .a py file. The .py file contains this line *oDesign.InitializeSystemCoupling("Optimetrics:SystemCouplingSetupn")* and must be preserved.

> **Note:**
>
> This API command is not captured when recording a script, but it can be used in a user-created script for automation.

UI Access	Right click a SystemCouplingSetup*n* nested under **Optimetrics** in the Project Manager Tree, and select **Generate Configuration Files**
Parameters	Name | Type | Description <*SystemCouplingSetup*> | String | Name of the Optimetrics setup to be used in system coupling.
Return 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	InitializeSystemCoupling(<*SystemCouplingSetup*>)
Python Example	`InitializeSystemCoupling("Optimetrics:SystemCouplingSetup1")`

InsertDesign

Inserts a new design in the project. For Icepak scripts, the last argument will always be empty.

UI Access	**Project** > **[Insert Icepak Design]**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignType*></td><td>String</td><td>Design type."Icepak".</td></tr><tr><td><*DesignName*></td><td>String</td><td>Design name.</td></tr><tr><td><SolutionType><ProblemType></td><td>String</td><td>Can be "SteadyState TemperatureAndFlow", "SteadyState TemperatureOnly", "SteadyState FlowOnly", "Transient TemperatureAndFlow", "Transient TemperatureOnly", or "Transient FlowOnly" .</td></tr><tr><td>""</td><td>None</td><td>Empty argument.</td></tr></table>
Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*SolutionType*>,'')
Python Example	`oProject.InsertDesign('Icepak','MyDesign', 'TemperatureAndFlow', '')`

PasteDesign (Design Object)

Pastes a design that has already been copied to the clipboard into another 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	**Edit** > **Paste**.		
Parameters	Name	Type	Description
	<*pasteOption*>	Integer	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')`
	`oDesign.PasteDesign(1)`

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()`

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	Name / Type / Description *<OldName>* / String / The current name of the design, which must be the design on which this command is invoked. *<NewName>* / String / The new name for the design.
Return Value	None.

Python Syntax	RenameDesignInstance (<*OldName*>, <*NewName*>)
Python Example	`oDesign.RenameDesignInstance("Design1", "Design2")`

RenameSource [Interface Source]

Use: Renames an interface source

Syntax: RenameSource (STRING: Interface Source 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.RenameSource "OldName" "NewName"`

RevertAllToInitialCondition

Reverts all setups to their initial condition. Used for linked thermal designs to revert the target solution (clearing restart, thermal monitor, and field results).

UI Access	**Project Manager** > right-click **Analysis** > **Revert to Initial Condition**. or, with either *nothing* or with **Analysis** selected in the Project Manager: **Icepak** > **Analysis** > **Revert to Initial Condition** (from the menu bar)
Parameters	None.
Return Value	None

Python Syntax	RevertAllToInitialCondition()
Python Example	`oModule.RevertAllToInitialCondition()`

RunCTMAPI

Runs a Python script, applying it to the active design. Running this command generates a matrix of resistance values that can be used to create a Network boundary condition. The script must have inputs defined based on files used for and generated by parametric trials.

The following files are required to run the RunCTMAPI command:

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

- **hfr.csv**: Heat flow rate values generated from parametric trials. The file contains N_e columns like the number of external nodes, and T rows like the number of trials.
- **areas.csv**: Sheet shape (area) data. The file contains N_e rows.
- **htc.csv**: Heat transfer coefficient values used as a boundary condition in the parametric trials. The file contains N_e columns like the number of external nodes and T rows like the number of trials.
- **power.csv**: Power values used as thermal load in the parametric trials. The file contains N_i columns like the number of internal nodes and T rows like the number of trials.
- **t_amb.csv**: Ambient temperature used in parametric trials. The file contains T rows.
- **temperature.csv**: Temperature values generated from parametric trials. The file contains N columns like the number of nodes (external + internal) and T rows like the number of trials
- **settings.toml**: Algorithm and optimizer settings. See "Settings.toml File" on page 7-63 for more information.

UI Access	[**None**
Parameters	Name | Type | Description *<RunCTMAPI>* | String | Generates a resistance value matrix.
Return Value	Run the *print(output)* command to display the resistance value matrix.

Python Syntax	RunCTMAPI(json_filename, hfr_filename, area_filename, htc_filename, power_filename, t_amb_filename, temperature_filename)

Python Example	`oDesign = oDesktop.GetActiveProject().GetActiveDesign()`
	
	`hfr_filename = r'C:\filepath\hfr.csv'`
	`area_filename = r'C:\filepath\areas.csv'`
	`htc_filename = r'C:\filepath\htc.csv'`
	`power_filename = r'C:\filepath\power.csv'`
	`t_amb_filename = r'C:\filepath\t_amb.csv'`
	`temperature_filename = r'C:\filepath\temperature.csv'`
	`json_filename = r'C:\filepath\settings.toml'`

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

	output = oDesign.RunCTMAPI(json_filename, hfr_filename, area_filename, htc_filename, power_filename, t_amb_filename, temperature_filename) print(output)

Note: In the example above, *filepath* denotes the location of the input files.

Note: You must include the print(output) command to display the resistance value matrix.

The following image displays the resistance value matrix output in the command prompt.

```
>>> output
['1.19586144280663', '-1.59091475548934E-08', '-1.96755879453558E-05', '-1.195839365412 58', '-2.91617381353859E-08', '-
5.21627185889884E-12', '-2.25939777998008E-09', '-1.15784051901713E-07', '-2.11066790045678E-07', '-1.08216984595622E-06', '-
9.45449913913876E-07', '-1.59091475548934E-08', '0.8450492171435 39', '-0.841930756254962', '-8.86064555061239E-11', '-
6.80750769088689E-08', '-0.000386538573681205', '-1.03357732506026E-06', '-0.00269467262063428', '-1.482642582195 66E-05', '-
2.13050567174911E-05', '-5.61566126933144E-10', '-1.96755879453558E-05', '-0.841930756254962', '0.842471631524041', '-
6.65023591750469E-13', '-4.5039118496959E-06', '-0.000153706451929247', '-9.39248678832882E-14', '-1.65334326229782E-08', '-
4.17940094199487E-06', '-7.48107242820595E-09', '-0.000358785901148462', '-1.19583936541258', '-8.86064555061239E-11', '-
6.65023591750469E-13', '1.31328460293934', '-0.116745030961652', '-2.62264201245088E-07', '-1.50752743621752E-11', '-
0.0001714776025230 07', '-0.000506387290066579', '-2.20792602718678E-05', '-4.37039293643693E-11', '-2.91617381353859E-08', '-
6.80750769088689E-08', '-4.5039118496959E-06', '-0.116745030961652', '0.118424425758433', '-0.00155308462097969', '-
4.14991284780619E-06', '-0.000100640716026001', '-8.31075053220331E-09', '-1.69093191382252E-05', '-7.6837403106822E-10', '-
5.21627185889884E-12', '-0.000386538573681205', '-0.000153706451929247', '-2.62264201245088E-07', '-0.00155308462097969',
'0.00435869638111219', '-1.75819359071738E-10', '-2.00007777007016E-08', '-4.82819099456577E-06', '-4.70009726205589E-06', '-
0.00225555600025085', '-2.25939777998008E-09', '-1.03357732506026E-06', '-9.39248678832882E-14', '-1.50752743621752E-11', '-
4.14991284780619E-06', '-1.75819359071738E-10', '0.024150644927893', '-0.0240278058513295', '-2.48984480588632E-07', '-
0.0001174040129805987', '-2.17177387185075E-11', '-1.15784051901713E-07', '-0.00269467262063428', '-1.65334326229782E-08', '-
0.0001714776025230 07', '-0.000100640716026001', '-2.00007777007016E-08', '-0.0240278058513295', '0.0318849999038022', '-
8.30300487209001E-06', '-0.00486345171803682', '-1.84960721183458E-05', '-2.11066790045678E-07', '-1.48264258219566E-05', '-
4.17940094199487E-06', '-0.000506387290066579', '-8.31075053220331E-09', '-4.82819099456577E-06', '-2.48984480588632E-07', '-
8.30300487209001E-06', '0.00110144948861368', '-7.04146838614861E-08', '-0.000562386399211467', '-1.08216984595622E-06', '-
2.13050567174911E-05', '-7.48107242820595E-09', '-2.20792602718678E-05', '-1.69093191382252E-05', '-4.70009726205589E-06', '-
0.0001174040129805987', '-0.00486345171803682', '-7.04146838614861E-08', '0.00573394900153956', '-0.000686939354704874', '-
9.45449913913876E-07', '-5.61566126933144E-10', '-0.000358785901148462', '-4.37039293643693E-11', '-7.6837403106822E-10', '-
0.00225555600025085', '-2.17177387185075E-11', '-1.84960721183458E-05', '-0.000562386399211467', '-0.000686939354704874',
'0.00388311057270974']
>>>
```


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

Settings.toml File

The following is an example of a Settings.toml file and descriptions of select settings.

```
[General]

cores = 8
it_shared = 250
max_gen = 500
pop = 230
pop_shared = 3
target_score = 0.99

[Operations]

deterministic = 5
mutation = 30
mutation_and_recombination = 60
recombination_crossover = 20
uniform_crossover = 20

[Elitism]

crossover_genes_fraction = 0.75
mutation_step_size = 0.2
recombination_expansion_parameter = 1.25
selection_pressure = 0.25

[Genes]

max_res = 1000000
max_res_final = 10000
min_res = 0.0001

[Scoring]

hfr_w = [0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01]
node_w = [
```


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

```
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
 0.09090909090909091,
]
temp_hfr_w = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
temp_w = [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]
```

General Settings

- **Population size**: Number of individuals in the population. More nodes in the network require more individuals. A larger number of individuals increases the optimization time.
- **Maximum generation**: Maximum number of generations (iterations) of the genetic algorithm. The optimization stops after this limit, regardless of the final score.
- **Target score**: The simulation stops if the specified target score is reached.
- **Node weights**: Allows the user to provide a comma-separated list of values to specify whether to overweight or underweight the accuracy of each node. The order of the nodes, whose weight is specified, is given with a tooltip.
- **Cores**: Number of cores to use in the genetic algorithm. Each core has its own population evolution and shares the best individuals after a certain number of iterations.
- **Individuals shared**: Number of individuals that each core shares with others after a certain number of iterations.
- **Iterations before sharing**: Number of iterations before a core shares its individuals.

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

Operation Settings

The sum of the number of children produced by all operations must be lower than the number of individuals in the population. In order to keep a good genetic diversity, it is suggested to keep around 30-50% of parents.

- **Mutation**: Number of children that have to be generated from a mutation operation. Mutation introduces random changes to an individual's genetic code to maintain genetic diversity.
- **Uniform Crossover**: Number of children that have to be generated from a uniform crossover operation. Uniform crossover creates children by randomly mixing genes from two parents.
- **Recombination crossover**: Number of children that are generated from a recombination crossover operation. Recombination crossover produces children by combining all genes from two parents using a random scale factor.
- **Mutation and recombination**: Number of children that are generated from a mutation-recombination operation. Mutation and recombination operation produces children by combining genes from two parents using a random scale factor.

Elitism Settings

- **Selection pressure**: Degree to which the best individuals are favored during selection. Higher selection pressure increases the likelihood that better individuals are chosen for the next generation, but on the other hand the genetic diversity is lowered.
- **Mutation step size**: The magnitude of changes introduced during mutation. Larger step sizes can result in more significant genetic variations.
- **Recombination expansion parameter**: Increase the scale factor in the mutation and recombination mutation.
- **Crossover genes fraction**: Controls the number of genes exchanged during recombination.

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.

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

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	[**Icepak**] > **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.

Return Value	User-defined solution(s) and/or output(s).

Python Syntax	RunToolkit(<*LibraryType*>, <*ToolkitName*>, <*ToolkitArg*>)
Python Example	`oDesign.RunToolkit("SysLib", "Geometry/Heatsinks/AngledFinHS", [])`

SARSetup

Sets up for the specific absorption rate (SAR) computation. This command does not apply to HFSS-IE.

UI Access	**HFSS** > **Fields** > **SAR Setting**.
Parameters	

Name	Type	Description
<*TissueMass*>	Double	Value represents mass of tissue between 1 and 10 in grams.
<*MaterialDensity*>	Double	Density of material in gram/cm^3.

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

	<*VoxelSize*>	Double	Size of a voxel in millimeters.
	<*AverageSARMethod*>	Integer	**0** - IEEE std 1528. **1** - Gridless, i.e. classical Ansoft method.
Return Value	None.		

Python Syntax	SARSetup(<*TissueMass*>, <*MaterialDensity*>, <*TissueObjectListID*>, <*VoxelSize*>, <*AverageSARMethod*>)
Python Example	`oDesign.SARSetup(1.0, 1.0, 678, 1.0, 0)`

SelectObjects

Selects and highlights the specified objects.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	SelectObjects()
Python Example	`oDesign.SelectObjects("[34,62]")`

SetActiveEditor

Sets the active 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	N/A.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*EditorName*></td><td>String</td><td>Text of the design's notes.</td></tr></table>
Return Value	Editor object.

Python Syntax	SetActiveEditor(<*EditorName*>)
Python Example	`oDesign.SetActiveEditor('3D Modeler')`

SetAllowMaterialOverride

Sets the option to allow material override.

UI Access	N/A.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*allowMaterialOverride*></td><td>Integer</td><td>**1** - allow material override. **0** - does not allow material override.</td></tr></table>
Return Value	None.

Python Syntax	SetAllowMaterialOverride(<*allowMaterialOverride*>)
Python Example	`oDesign.SetAllowMaterialOverride(1)`

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

SetBackgroundMaterial

Sets the design's background material.

> **Important:**
>
> You can only use this script in 2D Extractor if there is no surface ground in the design *and* the problem type is "open".

UI Access	From the **Project Manager**, right-click the design and select **Set Background Material**.		
Parameters	Name	Type	Description
	<*matName*>	String	Material name.
Return Value	None.		

Python Syntax	SetBackgroundMaterial(<*matName*>)
Python Example	`oDesign.SetBackgroundMaterial('vacuum')`

SetLengthSettings

Sets the distributed and lumped lengths of the design, as well as the rise time.

UI Access	N/A		
Parameters	Name	Type	Description
	<*DistributedUnits*>	String	Length units used for post-processing.
	<*LumpedLength*>	String	Length of design in post-processing.
	<*RiseTime*>	String	Time used in post-processing.
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	SetLengthSettings(<*DistributedUnits*>, <*LumpedLength*>, <*RiseTime*>)
Python Example	`oDesign.SetLengthSettings('mm', '7meter', '1s')`

SetPropValue [Design]

Sets the property value for the active property object.

UI Access	Edit properties on Project Tree objects.
Parameters	Name / Type / Description <*propPath*> / String / A child object's property path. See: Object Property Script Function Summary. <*Value*> / String / New property value.
Return Value	Boolean: • **True** - property found and the new value is valid. • **False** - property not found.

Python Syntax	SetPropValue(<*propPath*>, <*Value*>)
Python Example	`oDesign.SetPropValue("offset", "12mm")` `oDesign.SetPropValue("Results/S Parameter Plot 1/Display Type", "Rectangular Plot")`

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.

<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><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>
<tr><td><propName></td><td>String</td><td>Name of the property.</td></tr>
<tr><td><propValue></td><td>String</td><td>The value for the property</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	SetPropertyValue(<*propTab*>, <*propServer*>, <*propName*>, <*propValue*>)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

SetSolutionType

Sets the solution type for the design.

UI Access	**HFSS** > **Solution Type**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SolutionType*></td><td>String</td><td>Possible values are: "SBR+", "HFSS [Modal | Terminal] [Network | Composite]","Transient [Network | Composite]", "Eigenmode", or "Characteristic".</td></tr><tr><td>EnableAutoOpen:=,</td><td>Boolean</td><td>Only . • **True** - turn on auto open mode. • **False** - turn off auto open mode.</td></tr><tr><td><*ModelExteriorAsIE*></td><td>String</td><td>Only Applies for Driven Solution Type with Auto Open Mode as True. Possible values are: Radiation, FEBI, PML</td></tr></table>
Return Value	None.

Python Syntax	SetSolutionType(<*SolutionType*>, <*AutoOpenMode*>, <*ModelExteriorAsIE*>)
Python Example	`oDesign.SetSolutionType("HFSS Modal Network",`

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",
                "EnableAutoOpen:="      , False
        ])

oDesign.SetSolutionType("Transient Network",
        [
                "NAME:Options",
                "EnableAutoOpen:="      , False
        ])

oDesign.SetSolutionType("HFSS Hybrid Modal Network",
        [
                "NAME:Options",
                "EnableAutoOpen:="      , False
        ])

oDesign.SetSolutionType("SBR+",
        [
        "NAME:Options",
        "EnableAutoOpen:="      , False
```


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

	])

SetSolveInsideThreshold

Sets the solve inside threshold to the specified double.

UI Access	N/A		
Parameters	Name	Type	Description
	<threshold>	Double	Siemens/m.
Return Value	None.		

Python Syntax	SetSolveInsideThreshold(*<threshold>*)
Python Example	`oDesign.SetSolveInsideThreshold(100000)`

SetSourceContexts

For Near or Far Field projects for Driven Modal or Driven Terminal Network Analysis Solutions, specifies the port name and all modes/terminals of that port to be enabled as Source Context.

UI Access	**HFSS** > **Fields** > **Edit Sources**.		
Parameters	Name	Type	Description
	<SourceId>	Array	Name of modes/terminals to be set as source context.

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	SetSourceContexts(<*SourceId*>)

Python Example	`oModule.SetSourceContexts(`
	`[`
	`"Box1_T1", "Box1_T2", "Box1_T3", "Current1", "IncPWave1"`
	`])`

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.

ShowGeometry

Enables visualization for the specified object.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	oDesign.ShowGeometry()
Python Example	`oDesign.ShowGeometry("[34]")`

Undo [Design]

Cancels the last design-level command.

UI Access	**Edit** > **Undo**
Parameters	None.
Return Value	None.

Python Syntax	Undo()

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.Undo()`

ValidateDesign

Returns whether a design is valid.

UI Access	**Icepak** > **Validation Check**.
Parameters	None.
Return Value	Integer: • **1** – Validation passed. • **0** – Validation failed.

Python Syntax	ValidateDesign()
Python Example	`oDesign.ValidateDesign()`

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

CreateBondwire

CreateBox

CreateCircle

CreateCone

CreateCutplane

CreateCylinder

CreateEllipse

CreateEquationCurve

CreateEquationSurface

CreateHelix

CreatePoint

CreatePolyline

CreateRectangle

CreateRegion

CreateRegularPolygon

CreateRegularPolyhedron

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

CreateSphere

CreateSpiral

CreateSubregion

CreateTorus

CreateUserDefinedPart

Edit3DComponent

EditPolyline

EditNativeComponentDefinition [Beta – Layout Components in Icepak]

Get3DComponentDefinitionNames

Get3DComponentInstanceNames

Get3DComponentMaterialNames

Get3DComponentMaterialProperties

Get3DComponentParameters

Insert3DComponent

InsertNativeComponentDefinition [Beta – Layout Components in Icepak]

InsertPolylineSegment

SweepAlongPath

SweepAlongVector

SweepAroundAxis

SweepFacesAlongNormal

SweepFacesAlongNormalWithAttributes

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

UpdateComponentDefinition

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>,`
		`"XDir:=", <value>,`
		`"YDir:=", <value>,`
		`"ZDir:=", <value>,`
		`"Distance:=", <value>,`
		`"h1:=", <value>,`
		`"h2:=", <value>,`

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

			"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", "YPadPos:=" , "-0.2mm", "ZPadPos:=" , "0mm", "XDir:=" , "-2.2mm", "YDir:=" , "-1.4mm", "ZDir:=" , "0mm", "Distance:=" , "2.60768096208106mm", "h1:=" , "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.

```
  "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\"",
  "SurfaceMaterialValue:=", "\"\"",
  "SolveInside:="         , True,
  "ShellElement:="        , False,
  "ShellElementThickness:=", "0mm",
  "IsMaterialEditable:="  , True,
  "UseMaterialAppearance:=", False,
```


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

	"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>, "ZSize:=" , <string>]
	<AttributesArray>	Array	Structured array. See: AttributesArray.
Return Value	None.		

Python Syntax	CreateBox(*<Parameters>*, *<Attributes>*)
Python Example	oEditor.CreateBox(["NAME:BoxParameters",

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

```
    "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:="                , "",
    "MaterialValue:="        , "\"copper\"",
    "SurfaceMaterialValue:=" , "\"\"",
    "SolveInside:="          , False,
    "ShellElement:="         , False,
    "ShellElementThickness:=", "0mm",
    "IsMaterialEditable:="   , True,
```


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

	"UseMaterialAppearance:=", False, "IsLightweight:=" , False])

CreateCircle

Creates a circle.

UI Access	**Draw** > **Circle**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:CircleParameters", "IsCovered:=", <boolean>, "XCenter:=", <value>, "YCenter:=", <value>, "ZCenter:=", <value>, "Radius:=", <value>, "WhichAxis:=", <string> "NumSegments:=", <string containing integer>]</td></tr><tr><td><*AttributesArray*></td><td>Array</td><td>Structured array. See: AttributesArray.</td></tr></table>
Return Value	None.

Python Syntax	CreateCircle(<*Parameters*>, <*Attributes*>)

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: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)",
  "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])

CreateCone

Creates a cone.

UI Access	**Draw** > **Cone**.		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:ConeParameters", "XCenter:=", <string>, "YCenter:=", <string>, "ZCenter:=", <string>, "WhichAxis:=", <string>, "Height:=", <string>, "BottomRadius:=", <string>, "TopRadius:=", <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	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", "Name:=" , "Cone1", "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])

CreateCutplane

Creates a cutplane.

UI Access	**Draw** > **Plane**.		
Parameters	Name	Type	Description
	<*Parameters*>	Array	Structured array. ["NAME:PlaneParameters", "PlaneBaseX:=", <string>, "PlaneBaseY:=", <string>, "PlaneBaseZ:=", <string>, "PlaneNormalX:=", <string>, "PlaneNormalY:=", <string>,

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

			"PlaneNormalZ:=", <string>]
	<*AttributesArray*>	Array	See: AttributesArray. CreateCutplane only takes the Name and Color attributes.
Return Value	None.		

Python Syntax	CreateCutplane(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreateCutplane(["NAME:PlaneParameters", "PlaneBaseX:=" , "-0.6mm", "PlaneBaseY:=" , "-0.8mm", "PlaneBaseZ:=" , "0mm", "PlaneNormalX:=" , "1.2mm", "PlaneNormalY:=" , "0.2mm", "PlaneNormalZ:=" , "0mm"], ["NAME:Attributes", "Name:=" , "Plane1", "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.

CreateCylinder

Creates a cylinder.

<table>
<tr><td>UI Access</td><td>Draw > Cylinder.</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.
<pre>["NAME:CylinderParameters",
  "XCenter:=", <string>,
  "YCenter:=", <string>,
  "ZCenter:=", <string>,
  "Radius:=", <string>,
  "Height:=", <string>,
  "WhichAxis:=", <string>,
  "NumSides:=", <string containing integer>]</pre></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>None.</td></tr>
</table>

Python Syntax	CreateCylinder(<*Parameters*>, <*Attributes*>)

Python Example	`oEditor.CreateCylinder(`
	`["NAME:CylinderParameters",`
	`  "XCenter:="  , "6mm",`
	`  "YCenter:="  , "-4.5mm",`

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

```
    "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\"",
    "SurfaceMaterialValue:=", "\"\"",
    "SolveInside:="         , False,
    "ShellElement:="        , False,
    "ShellElementThickness:=", "0mm",
    "IsMaterialEditable:="  , True,
    "UseMaterialAppearance:=", False,
```


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

	"IsLightweight:=" , False])

CreateEllipse

Creates an ellipse.

UI Access	**Draw** > **Ellipse**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:EllipseParameters", "IsCovered:=", <string>, "XCenter:=", <string>, "YCenter:=", <string>, "ZCenter:=", <string>, "MajRadius:=", <string>, "Ratio:=", <string>, "WhichAxis:=", <string>, "NumSegments:=", <string>]</td></tr><tr><td><*AttributesArray*></td><td>Array</td><td>Structured array. See: AttributesArray.</td></tr></table>
Return Value	None.

Python Syntax	CreateEllipse(<*Parameters*>, <*Attributes*>)

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.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:="               , "",
  "Color:="               , "(143 175 143)",
  "Transparency:="        , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:="               , "",
  "MaterialValue:="       , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
```


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

```
  "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",`
			`  "XtFunction:=", <string>,`
			`  "YtFunction:=", <string>,`
			`  "ZtFunction:=", <string>,`
			`  "tStart:=", <string>,`
			`  "tEnd:=", <string>,`
			`  "NumOfPointsOnCurve:=", <string>,`
			`  "Version:=", <integer>),`

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

			<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, ["NAME:PolylineXSection", "XSectionType:=" , "None", "XSectionOrient:=" , "Auto", "XSectionWidth:=" , "0", "XSectionTopWidth:=" , "0", "XSectionHeight:=" , "0",

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

```
    "XSectionNumSegments:=" , "0",
    "XSectionBendType:="    , "Corner"
  ]
],
["NAME:Attributes",
  "Name:="                , "EquationCurve1",
  "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.

CreateEquationSurface

Creates an equation-based surface.

<table>
<tr><td>UI Access</td><td colspan="3">Draw > Equation-Based Surface.</td></tr>
<tr><td rowspan="2">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>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>,

"Version:=" , <integer>]</td></tr>
<tr><td></td><td><AttributesArray></td><td>Array</td><td>Structured array. See: AttributesArray.</td></tr>
<tr><td>Return Value</td><td colspan="3">None.</td></tr>
</table>

Python Syntax	CreateEquationSurface(<*Parameters*>, <*Attributes*>)

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.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)",
  "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])

CreateHelix

Creates a helix based on a sweep of specified objects.

UI Access	**Draw** > **Helix**.
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><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:HelixParameters", "XCenter:=" , <string>, "YCenter:=" , <string>, "ZCenter:=" , <string>, "XStartDir:=" , <string>, "YStartDir:=" , <string>, "ZStartDir:=" , <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.

			"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", "XStartDir:=" , "0mm", "YStartDir:=" , "10000mm", "ZStartDir:=" , "0mm", "NumThread:=" , "1", "RightHand:=" , True,

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

	"RadiusIncrement:=" , "0mm", "Thread:=" , "1mm"])

CreatePoint

Creates a point.

UI Access	**Draw** > **Point**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Parameters*></td><td>Array</td><td>Structured array. ["NAME:PointParameters", "PointX:=", <value>, "PointY:=", <value>, "PointZ:=", <value>]</td></tr><tr><td><*AttributesArray*></td><td>Array</td><td>Structured array. See: AttributesArray. CreatePoint takes only the Name and Color attributes.</td></tr></table>
Return Value	None.

Python Syntax	CreatePoint(<*Parameters*>, <*Attributes*>)
Python Example	oEditor.CreatePoint(["NAME:PointParameters", "PointX:=" , "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.

	"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>, <OnePointArray>, ...]
	<*OnePointArray*>	Array	["NAME:PLPoint", "X:=", <value>, "Y:=", <value>,

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

			"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"], ["NAME:PLPoint", "X:=" , "-90000mm",

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

```
    "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",
    "XSectionWidth:="       , "0mm",
    "XSectionTopWidth:="    , "0mm",
    "XSectionHeight:="      , "0mm",
    "XSectionNumSegments:=" , "0",
```


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

```
  "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**.
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: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", "Width:=" , "20000mm", "Height:=" , "30000mm", "WhichAxis:=" , "Z"

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:="                , "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
])
```

CreateRegularPolygon

Creates a regular polygon.

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>Draw > Regular Polygon.</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:RegularPolygonParameters",
  "IsCovered:=" , <boolean>,
  "XCenter:=" , <string>,
  "YCenter:=" , <string>,
  "ZCenter:=" , <string>,
  "XStart:=" , <string>,
  "YStart:=" , <string>,
  "ZStart:=" , <string>,
  "NumSides:=" , <string containing number greater than 2>,
  "WhichAxis:=" , <string "X", "Y", or "Z"]</pre></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>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>CreateRegularPolygon(<Parameters>, <Attributes>)</td></tr>
<tr><td>Python Example</td><td>
<pre>oEditor.CreateRegularPolygon(
["NAME:RegularPolygonParameters",
  "IsCovered:="   , True,
  "XCenter:="   , "-70000mm",
  "YCenter:="   , "-100000mm",</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.

```
    "ZCenter:="             , "0mm",
    "XStart:="              , "-50000mm",
    "YStart:="              , "-80000mm",
    "ZStart:="              , "0mm",
    "NumSides:="            , "12",
    "WhichAxis:="           , "Z"
],
["NAME:Attributes",
    "Name:="                , "Polygon1",
    "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
])
```

CreateRegularPolyhedron

Creates a regular polyhedron.

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>Draw > Regular Polyhedron.</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:PolyhedronParameters",
  "XCenter:=" , <string>,
  "YCenter:=" , <string>,
  "ZCenter:=" , <string>,
  "XStart:=" , <string>,
  "YStart:=" , <string>,
  "ZStart:=" , <string>,
  "Height:=" , <string>,
  "NumSides:=" , <string containing number greater than 2>,
  "WhichAxis:=" , <string "X", "Y", or "Z">]</pre></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>None.</td></tr>
</table>

Python Syntax	CreateRegularPolyhedron(<*Parameters*>, <*Attributes*>)

Python Example	`oEditor.CreateRegularPolyhedron(`
	`["NAME:PolyhedronParameters",`
	`  "XCenter:="  , "40000mm",`
	`  "YCenter:="  , "-80000mm",`
	`  "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.

```
  "XStart:="              , "50000mm",
  "YStart:="              , "-70000mm",
  "ZStart:="              , "0mm",
  "Height:="              , "50000mm",
  "NumSides:="            , "8",
  "WhichAxis:="           , "Z"
],
["NAME:Attributes",
  "Name:="                , "RegularPolyhedron1",
  "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
])
```

CreateSpiral

Creates a spiral by sweeping the specified object(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>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	<pre>oEditor.CreateSpiral(
	["NAME:Selections",
	  "Selections:=" , "Polygon2",
	  "NewPartsModelFlag:=" , "Model"
	],
	["NAME:SpiralParameters",</pre>

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

```
  "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",`
			`  "XCenter:=" , <string>,`
			`  "YCenter:=" , <string>,`
			`  "ZCenter:=" , <string>,`

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

			"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)", "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])

CreateUserDefinedModel

Creates a user-defined model.

Note: This option can be used to create a UDM from an imported CAD model (**Modeler** >).

UI Access	**Draw** > **User-Defined Model** > [**Model**]		
Parameters	Name	Type	Description
	<Parameters>	Array	Structured array. ["NAME:UserDefinedModelParameters", <DefinitionArray>, <OptionsArray>, <GeometryParamsArray>, "DllName:=", <string filepath>, "Library:=", <string>, "Version:=", <string>

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

		"ConnectionID:=", <string>]
<*DefinitionArray*>	Array	Structured array containing string "NAME:Definition." It defines the Discovery model to be used for import.
<*OptionsArray*>	Array	Structured array containing string "NAME:Options." The inputs correspond to the parameters in the **Options** > **3D Modeler** > **Discovery Link** window.
<*GeometryParamsArray*>	Array	Structured array containing arrays for individual parameters: ["NAME:GeometryParameters", ["NAME:UDMParam", "Name:=" , <string>, "Value:=" , <string>, "PropType2:=" , <integer>, "PropFlag2:=" , <integer>]] Required UDM parameters depend on the UDM being created. To see which properties apply to a UDM, right-click the UDM in the Project Tree and select **Properties**. Then select the **Parameters** tab. PropType2 can be any of the following: • **0** – Property takes a string value. • **1** – Property is a menu option. • **2** – Property takes a number (integer or double). • **3** – Property takes a value (numbers, variables, or expressions). • **4** – Property is a file name. • **5** – Property corresponds to a check box. • **6** – Property specifies a 3D position. PropFlag2 can be any 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.

			• **0** – No flags • **1** – Read-only • **2** – Must be integer • **4** – Must be real • **8** – Hidden `PropFlag2` values can be combined. For example, a read-only property that must be an integer would take the value 3. A hidden property that must be real would take the value 12. These values are further described in the `UserDefinedPrimitiveStructures.h` file included with the installation under "...\ANSYS Inc\v261\AnsysEM\UserDefinedPrimitives\Examples\Headers"

Return Value	None

Python Syntax	CreateUserDefinedModel(<*Parameters*>)

	Import a Unigraphics model:
	`oEditor = oDesign.SetActiveEditor("3D Modeler")`
	`oEditor.CreateUserDefinedModel(`
	`  ["NAME:UserDefinedModelParameters",`
	`  ["NAME:Definition",`
	`  ["NAME:UDMParam",`
	`  "Name:=" , "GeometryFilePath",`
	`  "Value:=" , "\"D:\\Temp\\model1.prt\"",`
	`  ],`

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

```
      ["NAME:UDMParam",
        "Name:=" , "ProcessID",
        "Value:=" , "50584",
      ],
      ["NAME:UDMParam",
        "Name:=" , "CAD Plug-In",
        "Value:=" , "\"Unigraphics\"",
      ]
    ]
  ])
```

Import a Discovery model with Options set:

```
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateUserDefinedModel(
  ["NAME:UserDefinedModelParameters",
    ["NAME:Definition",
      ["NAME:UDMParam",
        "Name:=" , "GeometryFilePath",
        "Value:=" , "\"D:/Designs/discovery_with_assembly 1.dsco\"",
        "DataType:=" , "String",
        "PropType2:=" , 0,
        "PropFlag2:=" , 0
      ],
    ],
    ["NAME:Options",
      ["NAME:UDMParam",
        "Name:=" , "Solid Bodies",
        "Value:=" , "1",
```


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

```
        "DataType:=" , "Int",
        "PropType2:=" , 5,
        "PropFlag2:=" , 0
      ],
      ["NAME:UDMParam",
        "Name:=" , "Surface Bodies",
        "Value:=" , "1",
        "DataType:=" , "Int",
        "PropType2:=" , 5,
        "PropFlag2:=" , 0
      ],
      ["NAME:UDMParam",
        "Name:=" , "Named Selections",
        "Value:=" , "1",
        "DataType:=" , "Int",
        "PropType2:=" , 5,
        "PropFlag2:=" , 0
      ],
      ["NAME:UDMParam",
        "Name:=" , "Parameters",
        "Value:=" , "\"eCADImportParameterType_Independ-
        ent,eCADImportParameterType_None,eCADImportParameterType_Independ-
        ent,eCADImportParameterType_All\"",
        "DataType:=" , "String",
        "PropType2:=" , 1,
        "PropFlag2:=" , 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:UDMParam",
          "Name:=" , "Parameter Key",
          "Value:=" , "\"\"",
          "DataType:=" , "String",
          "PropType2:=" , 0,
          "PropFlag2:=" , 0
        ],
        ["NAME:UDMParam",
          "Name:=" , "Materials",
          "Value:=" , "\"Assignments and Properties,None,Assignments,Assignments
          and Properties\"",
          "DataType:=" , "String",
          "PropType2:=" , 1,
          "PropFlag2:=" , 0
        ],
        ["NAME:UDMParam",
          "Name:=" , "Import As Lightweight",
          "Value:=" , "0",
          "DataType:=" , "Int",
          "PropType2:=" , 5,
          "PropFlag2:=" , 1
        ],
        ["NAME:UDMParam",
          "Name:=" , "Facet Level",
          "Value:=" , "\"3,1,2,3,4,5\"",
          "DataType:=" , "String",
          "PropType2:=" , 1,
          "PropFlag2:=" , 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:UDMParam",
          "Name:=" , "Cleaning",
          "Value:=" , "1",
          "DataType:=" , "Int",
          "PropType2:=" , 5,
          "PropFlag2:=" , 0
        ],
        ["NAME:UDMParam",
          "Name:=" , "Create Groups for Sub assembly",
          "Value:=" , "1",
          "DataType:=" , "Int",
          "PropType2:=" , 5,
          "PropFlag2:=" , 0
        ],
      ],
      ["NAME:GeometryParams"
      ],
      "DllName:=" , "SACADIntegUDM",
      "Library:=" , "installLib",
      "Version:=" , "1.0",
      "ConnectionID:=" , ""
  ])
```

Python Example

With GeometryParams set:

```
])
```


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

CreateUserDefinedPart

Creates a user-defined part.

<table>
<tr><td>UI Access</td><td colspan="3">Draw > User-Defined Primitive > [Part]</td></tr>
<tr><td rowspan="4">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Parameters></td><td>Array</td><td>Structured array.
["NAME:UserDefinedPrimitiveParameters",
"DllName:=" , <string>,
"Version:=" , <string>,
"NoOfParameters:=" , <integer>,
"Library:=" , <string>,
<paramVectorArray>]</td></tr>
<tr><td><paramVectorArray></td><td>Array</td><td>Structured array containing arrays for each pair:
["NAME:ParamVector",
<pair>, <pair>, <pair>,...
]
<pair>:
["NAME:Pair",
"Name:=" , <string>,
"Value:=" , <string>]</td></tr>
<tr><td><Attributes></td><td>Array</td><td>Structured array. See: AttributesArray.</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	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"], ["NAME:Pair", "Name:=" , "DiaYoke", "Value:=" , "20mm"], ["NAME:Pair", "Name:=" , "Length", "Value:=" , "100mm"], ["NAME:Pair", "Name:=" , "Skew", "Value:=" , "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:Pair",
"Name:=" , "Slots",
"Value:=" , "18"
],
["NAME:Pair",
"Name:=" , "SlotType",
"Value:=" , "1"
],
["NAME:Pair",
"Name:=" , "Hs0",
"Value:=" , "1mm"
],
["NAME:Pair",
"Name:=" , "Hs1",
"Value:=" , "1mm"
],
["NAME:Pair",
"Name:=" , "Hs2",
"Value:=" , "10mm"
],
["NAME:Pair",
"Name:=" , "Bs0",
"Value:=" , "2.5mm"
],
["NAME:Pair",
"Name:=" , "Bs1",
"Value:=" , "8mm"
],
```


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:=" , "Bs2",
        "Value:=" , "5mm"
        ],
        ["NAME:Pair",
        "Name:=" , "Rs",
        "Value:=" , "0mm"
        ],
        ["NAME:Pair",
        "Name:=" , "FilletType",
        "Value:=" , "0"
        ],
        ["NAME:Pair",
        "Name:=" , "Layers",
        "Value:=" , "2"
        ],
        ["NAME:Pair",
        "Name:=" , "CoilPitch",
        "Value:=" , "4"
        ],
        ["NAME:Pair",
        "Name:=" , "EndExt",
        "Value:=" , "5mm"
        ],
        ["NAME:Pair",
        "Name:=" , "SpanExt",
        "Value:=" , "25mm"
```


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:=" , "BendAngle",
    "Value:=" , "0deg"
    ],
    ["NAME:Pair",
    "Name:=" , "SegAngle",
    "Value:=" , "10deg"
    ],
    ["NAME:Pair",
    "Name:=" , "LenRegion",
    "Value:=" , "200mm"
    ],
    ["NAME:Pair",
    "Name:=" , "InfoCoil",
    "Value:=" , "0"
    ]
  ]
],
["NAME:Attributes",
  "Name:=" , "LapCoil1",
  "Flags:=" , "",
  "Color:=" , "(143 175 143)",
  "Transparency:=" , 0,
  "PartCoordinateSystem:=", "Global",
  "UDMId:=" , "",
  "MaterialValue:=" , "\"copper\"",
  "SurfaceMaterialValue:=", "\"\"",
```


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

	"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		
Parameters	Name	Type	Description
	<*compName*>	String	Component name.
	<*Parameters*>	Array	Structured array. ["NAME:EditComponentParametersData", "NewComponentName:=", <string>, "GeometryParameters:=", <string>, "MaterialParameters:=", <string>, "DesignParameters:=", <string>, <ComponentMeshing>, <Excitations>]

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

	<*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",`
	`"NewComponentName:=", "Connector2",`
	`"GeometryParameters:=", "",`
	`"MaterialParameters:=", "",`
	`"DesignParameters:=", "",`
	`["NAME:Component Meshing",`
	`"MeshAssembly:=", False],`
	`["NAME:Excitations",`
	`"Suppressed:=", []]`
	`]`

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

	)

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>,`
		`"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>]`

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	Edit3DComponent (<*Parameters*>)

Python Example	`oEditor.Edit3DComponent(`
	`  ["NAME:EditComponentParametersData",`
	`  "OriginalComponentName:=", "Connector1",`
	`  "NewComponentName:=", "Connector2",`
	`  "GeometryParameters:=", "",`
	`  "MaterialParameters:=", "",`
	`  "DesignParameters:=", "",`
	`  ["NAME:Component Meshing",`
	`  "MeshAssembly:=", False],`
	`  ["NAME:Excitations",`
	`  "Suppressed:=", []]`
	`  ]`
	`)`

EditNativeComponentDefinition [Beta – Layout Components in Icepak]

Use this command to edit the definition of an existing Layout Component in a design.

UI Access	In the Project Manager, expand **3D Components**, right-click on the component, and select **Edit Definition.**.
Parameters	**Name** / **Type** / **Description** *<LinkedParameters>* / Array / Structured array containing data of inserted layout 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	EditNativeComponentDefinition (<*LinkedParameters*>)

Python Example	`oEditor.EditNativeComponentDefinition(`
	`[`
	`"NAME:EditNativeComponentDefinitionData",`
	`"DefinitionName:=" , "LC1",`
	`[`
	`"NAME:GeometryDefinitionParameters",`
	`[`
	`"NAME:VariableOrders"`
	`]`
	`],`
	`[`
	`"NAME:DesignDefinitionParameters",`
	`[`
	`"NAME:VariableOrders"`
	`]`
	`],`
	`[`
	`"NAME:MaterialDefinitionParameters",`
	`[`
	`"NAME:VariableOrders"`
	`]`
	`],`
	`"NextUniqueID:=" , 0,`
	`"MoveBackwards:=" , False,`
	`"DatasetType:=" , "ComponentDatasetType",`

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

```
        [
                "NAME:DatasetDefinitions"
        ],
        [
                "NAME:NativeComponentDefinitionProvider",
                "Type:="                , "Layout Component",
                "Unit:="                , "mm",
                "Version:="             , 1.1,
                "EDBDefinition:="       , "SIwave-DC_Ice2",
        [
                "NAME:VariableMap"
        ],
                "ReferenceCS:="         , "",
                "CSToImport:="          , [],
                "BoardCutoutMaterial:=" , "air",
                "ViaHoleMaterial:="     , "copper",
                "ExtentsType:="         , "Bounding Box",
                "SliderResolution:="    , 2,
                "CustomResolution:="    , False,
                "UseThermalLink:="      , False,
                "Power:="               , "5W"
        ],
                "ComponentName:="       , "LC1",
                "Company:="             , "",
                "Company URL:="         , "",
                "Model Number:="        , "",
                "Help URL:="            , "",
                "Version:="             , "1.0",
                "Notes:="               , "",
                "IconType:="            , "Layout Component"
```


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

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>]`

<PolylineSegmentsArray>	Array	`<PolylineSegmentsArray>`
		`["NAME:PolylineSegments",`
		`<OneSegmentArray>, <OneSegmentArray>, ...]`

<OneSegmentArray>	Array	`["NAME:PLSegment",`

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

			"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"], ["NAME:PLPoint", "X:=" , "-90000mm",

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

```
    "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
  ]
```


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:="       , "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.

Python Syntax	Get3DComponentDefinitionNames()
Python Example	`oEditor.Get3DComponentDefinitionNames()`

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

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.

Python Syntax	Get3DComponentMaterialNames(<*InstanceName*>)

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.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.

Python Syntax	Get3DComponentParameters(<*compName*>)

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.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",`
	`"TargetCS:=", <string>,`
	`"ComponentFile:=", <string filepath>),`

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

			"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\' GuideWidth=\'0.9in\' WallThickness=\'0.1in\'", "MaterialParameters:=" , "",

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

	"DesignParameters:=" , ""], "XOffset:=" , "-70mm", "YOffset:=" , "45mm", "ZOffset:=" , "0mm"])

InsertComponent

Inserts a component.

UI Access	N/A
Parameters	Name: <*ComponentData*> — Type: Array — Description: Structured array. ["NAME:InsertComponentData", "Parameters:=", <string>, "TargetCS:=", <string>, "ComponentFile:=", <string filepath>]
Return Value	None.

Python Syntax	InsertComponent(<*ComponentData*>)
Python Example	oEditor.InsertComponent(

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

	["NAME:InsertComponentData", "Parameters:=", "", "TargetCS:=", "Global", "ComponentFile:=", "C:\tmp\Connector.a3dcomp"])

InsertNativeComponent [Beta – Layout Component in Icepak]

Use this command to insert a Layout Component into a design.

Note:

When the layout component being inserted into the design is not already present in the project, the *oEditor.InsertNativeComponent* command must be preceded by the *oComponentManager.Add* command to add the component to the project's *Components* Library. (See the **Thermal Solution** Python example below.

Similarly, when the source design contains geometry variables that you want to map to local design variables or AEDT project variables, one of the following commands must precede *oEditor.InsertNativeComponent*, depending on the preferred local variable type:

UI Access	In the Project Manager, right-click **3D Components** and select **Browse Layout Component**.
Parameters	**Name** / **Type** / **Description** <*LinkedParameters*> / Array / Structured array containing data of inserted layout component. See Python examples for details. Most parameters are self-explanatory. The following partial list includes parameters that need further explanation: <*MapInstanceParameters*> / String / Local variable type: "DesignVariable" or "ProjectVariable"

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

			(Applicable when geometry variables are mapped between the source design and target layout component)
	<*CustomResolution*>	Bool	Use slider position for trace mapping grid resolution when False Specify X & Y grid divisions numerically when True
	<*UseThermalLink*>	Bool	Import the power dissipation from a DCIR or AC analysis in the source HFSS 3D Layout design (applicable to *Steady-State* and *Transient* Icepak solutions)
	<*LayoutAnalysisType*>	String	Specify the type of solution in the source design for imported power dissipation (applicable when *UseThermalLink* is True. Choices are "DCIR" and "AC")
	<*SliderResolution*>	Integer	Slider position (tick mark number) used when <*CustomResolution*> = False (Controls resolution along the shorter board dimension: Values 0–4 represent resolutions of 50, 100, 200, 400, and 800, respectively)
	<*CustomResolutionCol*>	Integer	Number of columns (X resolution) when <*CustomResolution*> = True
	<*CustomResolutionRow*>	Integer	Number of rows (Y resolution) when <*CustomResolution*> = True
Return Value	None		

Python Syntax	InsertNativeComponent (<*LinkedParameters*>)
Python Example	`oComponentManager.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:TMT_03",
   "Info:=",
   [
      "Type:="            , 29,
      "NumTerminals:="    , 0,
      "DataSource:="      , "",
      "ModifiedOn:="      , 1711743004,
      "Manufacturer:="    , "",
      "Symbol:="          , "TMT_03",
      "ModelNames:="      , "",
      "Footprint:="       , "",
      "Description:="     , "",
      "InfoTopic:="       , "",
      "InfoHelpFile:="    , "",
      "IconFile:="        , "BlueDot.bmp",
      "Library:="         , "",
      "OriginalLocation:=", "Project",
      "IEEE:="            , "",
      "Author:="          , "",
      "OriginalAuthor:="  , "",
      "CreationDate:="    , 1711743004,
      "ExampleFile:="     , "",
      "HiddenComponent:=" , 0,
      "CircuitEnv:="      , 0,
      "GroupID:="         , 0
   ],
   "CircuitEnv:="         , 0,
   "Refbase:="            , "U",
   "NumParts:="           , 1,
```


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

```
        , True,
      "CompExtID:="          , 9,
      "ModelEDBFilePath:="   , "D:\\Ansys\\2024R2\\TMT_03.aedb",
      "EDBCompPassword:="    , ""
   ])
oDesign = oProject.SetActiveDesign("IcepakDesign2")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.InsertNativeComponent(
       [
              "NAME:InsertNativeComponentData",
              "TargetCS:="            , "Global",
              "SubmodelDefinitionName:=", "LC1",
              [
                     "NAME:ComponentPriorityLists"
              ],
              "NextUniqueID:="        , 0,
              "MoveBackwards:="       , False,
              "DatasetType:="         , "ComponentDatasetType",
              [
                     "NAME:DatasetDefinitions"
              ],
              [
                     "NAME:BasicComponentInfo",
                     "ComponentName:="       , "LC1",
                     "Company:="             , "",
                     "Company URL:="         , "",
                     "Model Number:="        , "",
                     "Help URL:="            , "",
                     "Version:="             , "1.0",
                     "Notes:="               , "",
```


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

	rmalLink:=" , True], ["NAME:InstanceParameters", "GeometryParameters:=" , "", "MaterialParameters:=" , "", "DesignParameters:=" , ""]])

InsertPolylineSegment

Insets a polyline segment before or after a specified existing segment.

UI Access	**Draw** > **Line Segment** > [**Selection**].
Parameters	See table below.
Return Value	None.

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.

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"])

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

SweepFacesAlongNormal

Sweep the specified face(s) along normal.

<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>Structured array.
["NAME:Parameters",
"NAME:SweepFaceAlongNormalToParameters",
"FacesToDetach:=", <faceIDarray>,
"LengthOfSweep:=", "<value><units>"]</td></tr>
</table></td></tr>
<tr><td>Return Value</td><td colspan="3">None</td></tr>
</table>

Python Syntax	SweepFacesAlongNormal(<*SelectionsArray*> <*parameters*>)
Python Example	oEditor.SweepFacesAlongNormal(["NAME:Selections", "Selections:=", "Rectangle1", "NewPartsModelFlag:=", "Model"], ["NAME:Parameters", "NAME:SweepFaceAlongNormalToParameters",

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

	"FacesToDetach:=", [183], "LengthOfSweep:=", "0.1mm"])

SweepFacesAlongNormalWithAttributes

Sweep a face along normal, and specify attributes of the new object.

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>["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>
Return Value	None

Python Syntax	SweepFacesAlongNormalWithAttributes(<*SelectionsArray*>, <*parameters*>, <*AttributesArray*>)
Python Example	oEditor.SweepFacesAlongNormalWithAttributes(["NAME:Selections", "Selections:=", "Rectangle1", "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:SweepFaceAlongNormalToParameters",
        "FacesToDetach:=", [183],
        "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])
```


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

UpdateComponentDefinition

Updates a 3D component's definition.

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

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

DeletePolylinePoint

DuplicateAlongLine

DuplicateAroundAxis

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([`

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"])

DeletePolylinePoint

Deletes either a start point or an end point from an existing polyline segment.

UI Access	**Edit** > **Delete** [**Start**/**End**] **Point**
Parameters	Name: *<DeletePointArray>* — Type: Array — Description: Structured array. ["NAME:Delete Point", "Selections:=", <string "<PolylineName>:<PolylineAction>:<int>">, "Segment Index:=", <integer>, "At Start:=", <bool True for start point; False for end point>]
Return Value	None.

Python Syntax	DeletePolylinePoint (*<DeletePointArray>*)
Python Example	oEditor.DeletePolylinePoint(["NAME:Delete Point", "Selections:=", "Polyline1:CreatePolyline:1", "Segment Index:=", 1, "At Start:=", True])

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

DuplicateAlongLine

Duplicates specified parts along a line.

<table>
<tr><td>UI Access</td><td colspan="3">Edit > Duplicate > Along Line.</td></tr>
<tr><td rowspan="5">Parameters</td><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>

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	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
	])

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

DuplicateAroundAxis

Duplicates specified parts around an axis.

<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.

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 > Mirror.</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:DuplicateToMirrorParameters",
"DuplicateMirrorBaseX:=", <string>,
"DuplicateMirrorBaseY:=", <string>,
"DuplicateMirrorNormalX:=", <string>,
"DuplicateMirrorNormalY:=", <string>,
"DuplicateMirrorNormalZ:=", <string>]</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	DuplicateMirror (<*SelectionsArray*>, <*ParametersArray*>, <*OptionsArray*>, <*CreateGroup*>)
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])

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

Mirror

Mirrors specified part(s).

<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"
	],</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:MirrorParameters", "MirrorBaseX:=" , "-0.2mm", "MirrorBaseY:=" , "-1.2mm", "MirrorBaseZ:=" , "0mm", "MirrorNormalX:=" , "-0.316227766016838mm", "MirrorNormalY:=" , "0.948683298050514mm", "MirrorNormalZ:=" , "0mm"])

Move

Moves specified part(s).

UI Access	**Edit** > **Arrange** > **Move**.
Parameters	Name / Type / Description <*SelectionsArray*> / Array / Structured array. See: SelectionsArray. <*TranslateParameters*> / Array / Structured array. ["NAME:TranslateParameters", "TranslateVectorX:=" , <string>, "TranslateVectorY:=" , <string>, "TranslateVectorZ:=" , <string>]
Return Value	None.

Python Syntax	Move(<*SelectionsArray*>, <*TranslateParameters*>)
Python Example	oEditor.Move(

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: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><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><*OffsetParameters*></td><td>Array</td><td>Structured array. ["NAME:OffsetParameters", "OffsetDistance:=" , <string>]</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	OffsetFaces (<*SelectionsArray*>, <*OffsetParameters*>)

Python Example	`oEditor.OffsetFaces(`
	`["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"`
	`])`

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

	`oEditor.Paste()`

Rotate

Rotates specified object(s).

UI Access	**Edit** > **Arrange** > **Rotate**.		
Parameters	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

	<*RotateParameters*>	Array	Structured array.
			`["NAME:RotateParameters",`
			`"RotateAxis:=" , <string "X", "Y", or "Z">,`
			`"RotateAngle:=" , <string>]`

Return Value	None.		

Python Syntax	Rotate(<*SelectionsArray*>, <*RotateParameters*>)

Python Example	`oEditor.Rotate(`
	`["NAME:Selections",`
	`"Selections:=", "Box1_1",`
	`"NewPartsModelFlag:=", "Model"`
	`],`
	`["NAME:RotateParameters",`

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

```
  "RotateAxis:=", "Z",
  "RotateAngle:=", "90deg"
])
```

Scale

Scales specified object(s).

UI Access	**Edit** > **Scale**.

Parameters

Name	Type	Description
<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.

<*ScaleParameters*>	Array	Structured array.
		`["NAME:ScaleParameters",`
		`  "ScaleX:=" , <string containing scale factor>,`
		`  "ScaleY:=" , <string containing scale factor>,`
		`  "ScaleZ:=" , <string containing scale factor>]`

Return Value	None.

Python Syntax	Scale (<*SelectionsArray*>, <*ScaleParameters*>)

Python Example

```
oEditor.Scale(
["NAME:Selections",
  "Selections:=", "Box1",
  "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:ScaleParameters", "ScaleX:=", "2", "ScaleY:=", "2", "ScaleZ:=", "2"])

Modeler Menu Commands

AssignMaterial

Chamfer

Connect

CoverLines

CoverSurfaces

CreateEntityList

CreateFaceCS

CreateGroup

CreateObjectCS

CreateObjectFromEdges

CreateObjectFromFaces

CreateRelativeCS

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

DeleteEmptyGroups

DeleteLastOperation

DetachFaces

EditEntityList

EditFaceCS

EditObjectCS

EditRelativeCS

Export

ExportModelImageToFile

ExportModelMeshToFile

Fillet

FlattenGroup

Generate History

HealObject

Import

ImportDXF

ImportGDSII [Modeler]

Intersect

MoveCStoEnd

MoveEntityToGroup

MoveFaces

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

ProjectSheet

PurgeHistory

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**

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><*argFaceList*></td><td>Array</td><td>["NAME:<SpecifiedName>", "BaseFaces:=" , <FaceNumberArray>, "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><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</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.

			• SolveInside • ShellElement • ShellElementThickness • 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`
	`])`

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

Chamfer

Creates a chamfer.

<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	`oEditor.Chamfer(`
	`["NAME:Selections",`
	`  "Selections:="  , "Box2",`

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:Parameters",
  ["NAME:ChamferParameters",
    "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.

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** > **Surface** > **Connect**.
Parameters	Name | Type | Description <*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",`

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

	"Selections:=", "Polyline3,Polyline4", "NewPartsModelFlag:=", "Model"])

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>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.
<pre>["NAME:GeometryEntityListParameters",
  "EntityType:=" , <string "Object" or "Face">,
  "EntityList:=" , <string of object names or IDs>]</pre>
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>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).

UI Access	**Modeler** > **Named Selection** > **Create** > [**Object** / **Face**] **Selection**

Parameters

Name	Type	Description

<Parameters>	Array	Structured array.
		`["NAME:NamedSelectionParameters",`
		`"Type:=" , <string "Object" or "Face">,`
		`"Selection:=" , <string of object names or IDs>]`
		See GetObjectIDByName for returning object IDs.

<AttributesArray>	Array	Structured array. See AttributesArray. CreateNamedSelection takes only the "Name" parameter and UDM ID.

Return Value	None.

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>Modeler > Coordinate System > Create > Object > [Offset / Rotated / Both].</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: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>
</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.

<table>
<tr><td>UI Access</td><td>Modeler > Edge > Create Object From Edge</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><ParametersArray></td><td>Array</td><td>Structured array.
["NAME:Parameters",
["NAME:BodyFromEdgeToParameters",
"Edges:=" , <array containing integer edges>
]
]</td></tr>
<tr><td><CreateGroupsForNewObjects></td><td>Array</td><td>Structured array.
["CreateGroupsForNewObjects:=", <boolean True to create groups for new objects; else False>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

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	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. See: SelectionsArray.
	<*Parameters*>	Array	Structured array. ["NAME:Parameters", <BodyFromFaceToParameters>]
	<*CreateGroupsForNewObjects*>	Array	Optional. Structured array. ["CreateGroupsForNewObjects:=", <boolean>]
Return Value	None.		

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.
["NAME:Parameters",
<DetachEdgesArray>]</td></tr>
<tr><td><DetachEdgesArray></td><td>Array</td><td>Structured array.
["NAME:DetachEdgesToParameters",
"EdgesToDetach:=" , <array containing integer edge IDs>]</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	<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", <DetachFacesArray>]</td></tr><tr><td><*DetachFacesArray*></td><td>Array</td><td>Structured array. ["NAME:DetachFacesToParameters", "FacesToDetach:=" , <array containing integer face IDs>]</td></tr></table>
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	Type	Description

	<*Parameters*>	Array	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 "Absolute-Position">, "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 colspan="3">Modeler > Import.</td></tr>
<tr><td rowspan="4">Parameters</td><td>Name</td><td>Type</td><td>Description</td></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>

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	Name	Type	Description
	<*SelectionsArray*>	Array	Structured array. List of objects to Boolean; see SelectionsArray.
	<*Parameters*>	Array	Structured array. ["NAME:IntersectParameters", "KeepOriginals:=" , <boolean True to keep original objects; False to delete>]
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	Type	Description
	<*SelectionsArray*>	Array	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	Type	Description
	<*Objects*>	Array	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.

<table>
<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:SectionToParameters",
"CreateNewObjects:=" , <boolean>,
"SectionPlane:=" , <string "XY", "YZ", or "ZX">,
"SectionCrossObject:=" , <boolean>]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

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	Type	Description
	<*SelectionsArray*>	Array	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><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:SplitToParameters",</code>

<code>"SplitPlane:=" , <string "XY", "YZ", "ZX", or "Dummy">,</code>

<code>"WhichSide:=" , <string "PositiveOnly", "NegativeOnly" or "Both">,</code>

<code>"ToolType:=" , <string "PlaneTool" or "FaceTool">,</code>

<code>"ToolEntityID:=" , <-1 if using SplitPlane; else FaceID>,</code>

<code>"SplitCrossingObjectsOnly:=", <boolean>,</code>

<code>"DeleteInvalidObjects:=", <boolean>]</code>

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 colspan="3">Modeler > Boolean > Subtract.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<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 colspan="3">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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Groups*></td><td>Array</td><td>Structured array. `["Groups:=", <array of group names to ungroup>]`</td></tr></table>
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	<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:WrapSheetParameters",` `  "Imprinted:=", <boolean>]`</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	WrapSheet (<*SelectionsArray*>, <*Parameters*>)

Python Example	`oEditor.WrapSheet(`
	`["NAME:Selections",`
	`  "Selections:=","Rectangle1,Box1"`
	`],`
	`["NAME:WrapSheetParameters",`
	`  "Imprinted:=", True`
	`]`
	`)`

Other oEditor Commands

AddDefinitionFromBlock

AddDefinitionFromLibFile

BreakUDMConnection

ChangeProperty

Delete

FitAll

GetBodyNamesByPosition

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

GetChildNames [Modeler]

GetChildOjbect [Modeler]

GetChildTypes [Modeler]

GetEdgeByPosition

GetEdgeIDsFromObject

GetEdgeIDsFromFace

GetEntityListIDByName

GetExtendedDefinitionObject

GetFaceArea

GetFaceByPosition

GetFaceCenter

GetFaceIDs

GetGeometryModelerMode

GetMatchedObjectName

GetModelBoundingBox

GetModelUnits

GetNumObjects

GetObjectAttributeValues

GetObjectIDByName

GetObjectName

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

GetObjectNameByFaceID

GetObjectsByMaterial

GetObjectsInGroup

GetObjectVolume

GetPropertyValue

GetPropEvaluatedValue

GetPropNames

GetPropSIValue

GetPropValue

GetSelections

GetUserPosition

GetVertexIDsFromEdge

GetVertexIDsFromFace

GetVertexIDsFromObject

GetVertexPosition

PageSetup

RenamePart

SetObjectAttributeValues

SetPropValue [Modeler]

Setting User-Defined Attributes

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",
    "ZPosition:=" , "0mm",
```


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

```
    "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 = ''
  if isinstance(added, basestring):
```


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

```
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
        ]
      ]
    ]
  ])
```

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

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 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(
  ["NAME:BoxParameters",
    "XPosition:=" , "-0.4mm",
    "YPosition:=" , "-1mm",
    "ZPosition:=" , "0mm",
    "XSize:=" , "1.4mm",
    "YSize:=" , "1.6mm",
    "ZSize:=" , "0.6mm"
  ],
```


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:=" , "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 + "\""
added = oDefinitionManager.AddDefinitionFromLibFile(amatFilePath, mater-
ialName, "Materials", "")
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)`
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DAttributeTab",`
	`  ["NAME:PropServers",`
	`  "Box1"`
	`  ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Material",`
	`  "Value:=", materialNameInQuotes`
	`  ]`
	`  ]`
	`  ]`
	])

AddViewOrientation

Creates a new orientation using a script. Input parameters can be *either* vector components *or* angles. Local view orientation specific to a design can be saved in project by calling Save() on the project object.

> **Note:**
>
> You cannot use the name of an existing view. Default views include: top, bottom, right, left, front, back, trimetric, dimetric, and isometric.

UI Access	**View** > **Modify Attributes** > **Orientation List**. Enter parameters and click **Add**.

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
<orientationName>	String	Name of the new orientation.
<isGlobalOrientation>	Long	1 to save global; else 0.

<OrientationVectorComponents>	Array	Structured array. Use *either* <OrientationVectorComponents> *or* <OrientationAnglesArray>.
		`["NAME:OrientationVectorComponents",`
		`  ["NAME:Direction",`
		`  <float>,`
		`  <float>,`
		`  <float>],`
		`  ["NAME:Up",`
		`  <float>,`
		`  <float>,`
		`  <float>]]`

<OrientationAnglesArray>	Array	Structured array. Use *either* <OrientationVectorComponents> *or* <OrientationAnglesArray>.
		`["NAME:OrientationAngles",`
		`  "Psi:=" , <float>,`
		`  "Theta:=" , <float>,`
		`  "Phi:=" , <float>]`

<ProjectionParams>	Array	*Optional*. Structured array containing arrays of four optional parameters: Scaling, Translation, Projection Type, and Projection Limits.

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

```
["NAME:ProjectionParams",
  ["NAME:Scaling",
    <float>,
    <float>,
    <float>
  ],
  ["NAME:Translation",
    <float>,
    <float>,
    <float>
  ],
  "ProjectionType:=" ,
  <string "Orthographic" or "Perspective">,
  ["NAME:ProjectionLimits",
    <float>,
    <float>,
    <float>,
    <float>,
    <float>,
    <float>
  ]
]
```


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	AddViewOrientation(<*orientationName*>, <*isGlobalOrientation*>, <*OrientationVectorComponents* or *OrientationAnglesArray*>, <*ProjectionParams*>)

Python Example	**Example Using Angles and Projection Parameters:**
	`oEditor.AddViewOrientation("orientation3", 0,`
	`  ["NAME:OrientationAngles",`
	`  "Psi:=" , -161.565002441406,`
	`  "Theta:=" , 32.3115005493164,`
	`  "Phi:=" , 0 ],`
	`  ["NAME:ProjectionParams",`
	`  ["NAME:Scaling",`
	`  0.591607987880707,`
	`  0.591607987880707,`
	`  0.591607928276062`
	`  ],`
	`  ["NAME:Translation",`
	`  0,`
	`  2.38418579101563E-07,`
	`  -5.7486834526062`
	`  ],`
	`  "ProjectionType:=" , "Orthographic",`
	`  ["NAME:ProjectionLimits",`

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

```
        -2.44362282752991,
        2.44362282752991,
        -1,
        1,
        0.625207424163818,
        10.8721580505371
     ]
   ]
)
```

Example Using Vector without Projection Parameters:

```
oEditor.AddViewOrientation("orientation6", 0,
["NAME:OrientationVectorComponents",
   ["NAME:Direction",
     -0.801783978939056,
     -0.267262011766434,
     -0.534521996974945
   ],
   ["NAME:Up",
     -0.507091999053955,
     -0.169030994176865,
     0.845155000686646
   ]
])
```

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.
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.

Return Value	None.

Python Syntax	BreakUDMConnection (<*SelectionsArray*>)

Python Example	`oEditor.BreakUDMConnection(`
	`[`
	`"NAME:Selections",`
	`"Selections:=" , "Discovery1"`
	`])`

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()`

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

Defeature

Removes irrelevant features from a primitive.

UI Access	N/A

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><*Options*></td><td>Array</td><td>Structured array.
	`["NAME:options",`
	`"tolerance:=", <double containing tolerance value>,`
	`"fix:=", <boolean, if true, then self-intersections are fixed.>]`</td></tr></table>

Return Value	None.

Python Syntax	Defeature(<*Selections*>, <*Options*>)

Python Example	`oEditor.Defeature(`
	`["NAME:Selections", "Selections:=", "Box1"],`
	`["NAME:Options", "Tolerance:=", 1E-006, "Fix:=", True])`

Delete

Deletes the specified object(s).

UI Access	**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.

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	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()`

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

GenerateAllUserDefinedModels

Generates all user defined models.

UI Access	N/A
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*>)

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.GenerateUserDefinedModel(["NAME:Selections", "Selections:=", "model1, model2"])`

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>)**

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"

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

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"
- "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 sys-</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.

			tem.
			`["NAME:Parameters",`
			`"XPosition:=", <string>,`
			`"YPosition:=", <string>,`
			`"ZPosition:=", <string>]`

Return Value	Array containing string object names.		

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.

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	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*category*></td><td>String</td><td>*Optional*. Passing no input returns the list of possible strings: "AllParts" – Returns the names of all parts. "CoordinateSystems" – Returns the names of all 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. </td></tr></table>
Return Value	Array containing object names in the selected category.

Python Syntax	GetChildNames(<*category*>)
Python Example	Standalone 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.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".

```
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']`

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

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.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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", "Coordin-

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

	ateSystems", "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()`
	• 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.

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	N/A
Parameters	None.
Return Value	Array containing the names of child types.

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()`

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 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.

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.

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	GetEdgeByPosition (<*positionParameters*>)

Python Example	`oEditor.GetEdgeByPosition(`
	`  ["NAME:EdgeParameters",`
	`  "BodyName:=", "Box1",`
	`  "Xposition:=", "10mm",`
	`  "YPosition:=", "0mm",`
	`  "ZPosition:=", "10mm"`
	`  ]`
	`)`

GetEdgeIDFromNameForFirstOperation

Gets edge ID from first operation of a part.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PartName*>	String	Name of specified part.
	<*EdgeName*>	String	Name of specified edge.
Return Value	Integer edge ID		

Python Syntax	GetEdgeIDFromNameForFirstOperation(<*PartName*>, <*EdgeName*>)

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.GetEdgeIDFromNameForFirstOperation("Coil_1", "Edge_4510")`

GetEdgeIDsFromFace

Returns the edge IDs for a specified Face ID.

UI Access	N/A
Parameters	Name | Type | Description <*faceID*> | Integer | 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 | Type | Description <*object*> | String | Object name.
Return Value	Array containing string edge 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	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
Parameters	Name | Type | Description <*selectionName*> | String | selection 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	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")`

GetExtendedDefinitionObject

Get an object-oriented property scripting object by name and library type (using definition folder name) which also supports getting/setting mod time and getting/setting generic string attributes. This scripting command directly supports the .AMAT (or .ASURF)

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

definition formats.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*defName*></td><td>String</td><td>Definition to retrieve.</td></tr><tr><td><*defFolderName*></td><td>String</td><td>Library type (by definition folder name).</td></tr></table>
Return Value	An extended material wrapper script object (see material wrapper script object functions above).

GetFaceArea

Returns the area of a specified face.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*faceID*></td><td>Integer</td><td>ID of specified face.</td></tr></table>
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.

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><*planarFaceID*></td><td>Long</td><td>Face ID</td></tr></table>
Return Value	Array containing string X, Y, Z coordinates.

Python Syntax	GetFaceCenter (<*planarFaceID*>)
Python Example	`oEditor.GetFaceCenter(12)`

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FaceParameters*></td><td>Array</td><td>Structured Array.
	`["NAME:FaceParameters",`
	`"BodyName:=", <string>,`
	`"XPosition:=", <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.

			"YPosition:=", <string>, "ZPosition:=", <string>]
Return Value	Integer Face ID.		

Python Syntax	GetFaceByPosition(<*FaceParameters*>)
Python Example	oEditor.GetFaceByPosition(["NAME:FaceParameters", "BodyName:=", "Box1", "XPosition:=", "0.2mm", "YPosition:=", "-0.2mm", "ZPosition:=", "0.4mm"])

GetFaceIDFromNameForFirstOperation

Gets face ID from first operation of a part.

UI Access	N/A		
Parameters	Name	Type	Description
	<*PartName*>	String	Name of specified part.
	<*FaceName*>	String	Name of specified face.
Return Value	Integer face 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	GetFaceIDFromNameForFirstOperation(<*PartName*>, <*FaceName*>)
Python Example	`oEditor.GetFaceIDFromNameForFirstOperation("Rotor", "Face_14343"))`

GetFaceIDs

Returns the face IDs associated with a specified object.

UI Access	N/A
Parameters	Name | Type | Description <*objectName*> | String | 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

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><*sheetName*></td><td>String</td><td>Name of specified sheet object.</td></tr></table>
Return Value	Array containing string face IDs.

Python Syntax	GetFaceIDsOfSheet(<*sheetName*>)
Python Example	`oEditor.GetFaceIDsOfSheet('Sheet1')`

GetGeometryModelerMode

Returns the modeler mode (3D or XY) for the current design. This lets you know whether the current model is 2D or 3D.

UI Access	N/A
Parameters	None.
Return Value	String containing the modeling mode ("3D" or "XY").

Python Syntax	GetGeometryModelerMode()
Python Example	`oEditor.GetGeometryModelerMode()`

GetGroupSubmodelNames

Returns name of models belong to a specific group.

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><*groupName*></td><td>String</td><td>Group name.</td></tr></table>
Return Value	Array of strings containing the model names.

Python Syntax	GetGroupSubmodelNames(<*groupName*>)
Python Example	`oEditor.GetGroupSubmodelNames("Group 1")`

GetMatchedObjectName

Returns all object names containing the input text string.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*wildcardText*></td><td>String</td><td>Text string present in object name(s).</td></tr></table>
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.
Return Value	String containing 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.

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 | Description <*FaceID*> | Integer | The face ID.
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)`

GetObjectsByMaterial

Returns a list of objects of a specified material.

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
	<*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*>)
Python Example	`oEditor.GetObjectShapeType("box1", "Global")`

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

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')`

GetObjectVolume

Returns an object's volume from its name).

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 <*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: <*udmName*> — Type: String — Description: 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	<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	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>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 [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")`
	return "length / 2"
	`oCreateBox.GetPropEvaluatedValue("xSize")`
	return '0.4mm'

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

	`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:
	`'Global'`

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

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

Python Syntax	GetSelections()
Python Example	`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.

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')`

GetUserPosition

Returns a user's current coordinates in the 3D Modeler window.

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
	<*prompt*>	String	"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	Type	Description
	<*PartName*>	String	Name of specified part.
	<*VertexName*>	String	Name of specified vertex.
Return Value	Integer representing vertex ID.		

Python Syntax	GetVertexIDFromNameForFirstOperation(<*PartName*>, <*VertexName*>)
Python Example	`oEditor.GetVertexIDFromNameForFirstOperation("Part1", "Vertex1")`

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

GetVertexIDsFromEdge

Returns vertex IDs associated with a specified edge.

UI Access	N/A
Parameters	Name: *<edgeID>*; Type: Integer; Description: 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: *<faceID>*; Type: Integer; Description: 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
Parameters	Name | Type | Description <*vertexID*> | Integer | Vertex ID.
Return Value	Array containing X, Y, Z coordinates.

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	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.

UI Access	**File** > **Page Setup**.
Parameters	Name | Type | Description <*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:PageSetupData", "margins:=", <SetupArray>]
	<*SetupArray*>	Array	Structured Array ["left:=", <string>, "right:=", <string>, "top:=", <string>, "bottom:=", <string>)]
Return Value	None.		

Python Syntax	PageSetup(<*Parameters*>)
Python Example	oEditor.PageSetup(["NAME:PageSetupData", "margins:=", ["left:=", "10mm", "right:=", "10mm", "top:=", "10mm", "bottom:=", "10mm"]

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

	])

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.
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	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.		
Parameters	Name	Type	Description
	<*renameParametersArray*>	Array	Structured array. `["NAME:Rename Data",`

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

			"Old Name:=", <string>, "New Name:=", <string>]
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.

> **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.

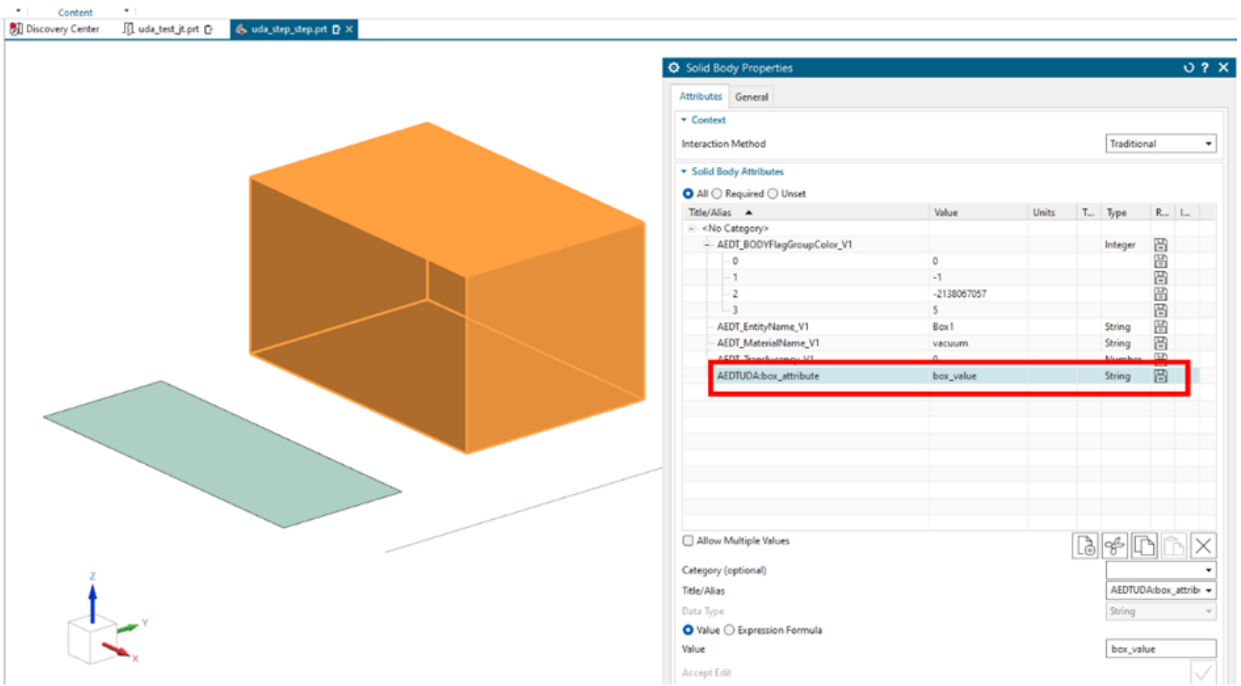
UI Access	Edit **Properties** on History Tree objects
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*propPath*></td><td>String</td><td>Child object's property path. See: Object Property Function Summary.</td></tr><tr><td><*value*></td><td>Varies</td><td>New property 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:

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

- SetObjectAttributeValues
- GetObjectAttributeValues
- RemoveObjectAttributeValues

SetObjectAttributeValues

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></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

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><*objectName*></td><td>String</td><td>Name of object to query</td></tr><tr><td><*Type*></td><td>String</td><td>Only possible value is "AEDTUDA"</td></tr><tr><td><*stringFilter*></td><td>String</td><td>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</td></tr></table>
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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*objectName*></td><td>String</td><td>Name of object that has the UDAs you want to remove</td></tr><tr><td><*Type*></td><td>String</td><td>Only possible value is "AEDTUDA"</td></tr><tr><td><*stringFilter*></td><td>String</td><td>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</td></tr></table>
Return Value	Returns a value of 1 if attribute is successfully deleted

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	RemoveObjectAttributeValues(<*objectName*>, "AEDTUDA", <*stringFilter*>)
Python Example	`oEditor = oDesign.SetActiveEditor("3D Modeler")` `oEditor.RemoveObjectAttributeValues("Box1", "AEDTUDA", [" "])`

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")`

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

SetTopDownViewDirectionForAllViews

Sets all views to top-down view direction.

UI Access	N/A
Parameters	Name: <*RelativeCS*> — Type: String — Description: 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: <*Lists*> — Type: Array — Description: Array of priority list names.
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	UpdatePriorityList (<*Lists*>)
Python Example	`oEditor.UpdatePriorityList(["PriorityList1", "PriorityList2"])`

UpgradeVersion

Upgrades legacy geometry to current version.

UI Access	Right-click on an operation icon in the history tree, select **Upgrade Version**.

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Selections*></td><td>Array</td><td>Structured array.
	`["NAME:Parameters",`
	`  ["NAME:PartOperations",`
	`  ["NAME:<string>",],`
	`  "OperationIndices:=", <array of integers>`
	`  ],`
	`  ["NAME:UDMOperations"]`
	`]`</td></tr></table>

Return Value	None.

Python Syntax	UpgradeVersion (<*Selections*>)

Python Example	`oEditor.UpgradeVersion([`
	`  "NAME:Parameters",`
	`  ["NAME:PartOperations",`
	`  ["NAME:source",`

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

	"OperationIndices:=", [0]]], ["NAME:UDMOperations"]])

Validate3DComponent

Validates a 3D component.

UI Access	N/A
Parameters	Name / Type / Description <*componentPath*> / String / Path to a 3D component. <password> / String / Optional. Password to access the component.
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","")

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

WriteHistoryTreeLayoutForTest

Writes history tree layout to a file.

UI Access	N/A
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

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

CreateTwistedPairCable

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, [

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

```
                        "NAME:InsulationJacketParams",
                        "InsThickness:="        , "0.25mm",
                        "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",
```


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

```
                        "tpwire2/w1:="          , [                             "Imped:="
                        "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(`
	`[`
	`"NAME:ClockSignalParams",`

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

```
                "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", ["NAME:VirtualJacketParams",

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

```
                "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")`

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.

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	**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>))
Python Example	`oModule.UpdateCableHarness(` `[`

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:="
        "tpwire2/w2:="          , [                     "Source:="
```


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

```
                ]
        ],
        [
                "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.

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:

CreateOutputVariable

DeleteOutputVariable

DoesOutputVariableExist

EditOutputVariable

GetOutputVariableValue

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.

UI Access	**Icepak** > **Results** > **Output Variables**.		
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.

	<OutputVarName>	String	Name of the new output variable.
	<Expression>	Value	Value to assign to the variable.
	<SolutionName>	String	Name of the solution, as seen in the output variable UI.
	<reportType>	String	The name of the report type as seen in the output variable UI.
	<ContextArray>	Array	Structured array containing context for which the output variable expression is being evaluated. `["Context:=", <string>]`
Return Value	None.		

Python Syntax	CreateOutputVariable (*<OutputVarName>*, *<Expression>*, *<SolutionName>*, *<reportType* | *solutionType>*, *<contextArray* | *domainArray>*)
Python Example	

DeleteOutputVariable

Deletes an existing output variable. The variable can only be deleted if it is not in use by any traces.

UI Access	**Icepak** > **Results** > **Output Variables**. In the **Output Variables** window, click **Delete**.		
Parameters	Name	Type	Description
	<OutputVarName>	String	Name of the output variable

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	DeleteOutputVariable (<*OutputVarName*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.DeleteOutputVariable ("testNew")`

DoesOutputVariableExist

Verifies whether or not a named output variable exists.

UI Access	N/A		
Parameters	Name	Type	Description
	<*outputVariableName*>	String	Output variable name
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")`

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

EditOutputVariable

Changes the name or expression of an existing output variable.

<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><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, <code>"Setup1 : Last Adaptive"</code>.</td></tr>
<tr><td><ReportType></td><td>String</td><td>Name of the report type as seen in the output variable UI</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated.

<code>["Context:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	EditOutputVariable (<*OrigVarName*>, <*NewExpression*>, <*NewVarName*>, <*SolutionName*>, <*ReportType*>, <*ContextArray*>)

Python Example	`oModule = oDesign.GetModule("OutputVariable")`
	
	`oModule.EditOutputVariable ("test", "normalize(R1_0.V)", "testNew", "TR", "Standard", [])`

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

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.
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.

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	GetOutputVariables()
Python Example	`oDesign.GetOutputVariables()`

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.

UI Access	N/A
Parameters	

Name	Type	Description
<*OutputVarName*>	String	Name of the output variable.
<*IntrinsicVariation*>	String	A set of intrinsic variable value pairs to use when evaluating the output expression.
<*SolutionName*>	String	Name of the solution as listed in the output variable UI. For example, `"Setup1 : Last Adaptive"`.
<*ReportType*>	String	The name of the report type as seen in the output variable UI.
<*ContextArray*>	Array	Structured array containing context for which the output variable expression is being evaluated. Can be empty. `["Context:=", <string>]`

Return Value	Double value of the output variable.

Python Syntax	GetOutputVariableValue(<*OutputVarName*>, <*IntrinsicVariation*>, <*SolutionName*>, <*ReportTypeName*>, <*ContextArray*>)
Python Example	`Val = oDesign.GetOutputVariableValue("test","Freq = '20Ghz' Theta='20deg'`

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

	`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")`

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,
```


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

```
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
NetworkFunction Domain	9
Oscillator Domain	55802
Noise Domain	55803
Transfer Function Domain	55807
Time Frequency Domain	55808

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

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
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.

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

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.

— **"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.

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

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
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

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

* 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:

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:

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
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:

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

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

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

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

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

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.

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.

10 - Reporter Editor Script Commands

Reporter commands should be executed by the oDesign object.

For example:

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:

AddCartesianLimitLine

AddCartesianXMarker

AddCartesianYMarker

AddCartesianYMarkerToStack

AddDeltaMarker

AddMarker

AddNote

AddTraces

AddVerifEyeAnalysis

ClearAllMarkers

ClearAllTraceCharacteristics

CopyReportDefinitions

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

CopyReportData

CopyTraceDefinitions

CopyTracesData

CreateReport

CreateReportFromFile

CreateReportFromTemplate

CreateReportOfAllQuantities

DeleteAllReports

DeleteReports

DeleteTraces

EditQuickEyeAnalysis

EditVerifEyeAnalysis

ExportImageToFile

ExportToFile [Reporter]

GetAllCategories

GetAllQuantities

GetAllReportNames

GetAvailableDisplayTypes

GetAvailableReportTypes

GetAvailableSolutions

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

GetChildNames

GetChildObject

GetChildTypes

GetPropertyValue

GetSolutionContexts

GetSolutionDataPerVariation

GroupPlotCurvesByGroupingStrategy

ImportIntoReport

MovePlotCurvesToGroup

MovePlotCurvesToNewGroup

PasteReports

PasteTraces

RenameReport

RenameTrace

ResetPlotSettings

SavePlotSettingsAsDefault

SetPropValue

UpdateAllFieldsPlots

UpdateAllReports

UpdateReports

UpdateTraces

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

UpdateTracesContextandSweeps

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>],`

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

	"XUnits:=", <string unit of measure for X>, ["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.

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

			["NAME:CartesianLimitLineFromCurve", "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",

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

```
  "YAxis:="               , 1,
  "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.

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	AddCartesianLimitLineFromEquation(<*ReportName*>, <*LimitLineParams*>)

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></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	
Return Value	None		

Python Syntax	AddDeltaMarker(<*ReportName*>, <*MarkerName1*>, <*CurveName1*>, <*PrimarySweepValue1*>, <*MarkerName2*>, <*CurveName2*>, <*PrimarySweepValue2*>)
Python Example	

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.
Return Value	None		

Python Syntax	AddMarker(<*ReportName*>, <*MarkerName*>, <*CurveName*>, <*PrimarySweepValue*>)
Python Example	`oModule.AddMarker("XY Plot 1", "m5",` `"GS1.VAL : TR4 : Cartesian", "3.61599999999997s")`

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

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>,`
		`"DefaultYPos:=", <int Y position for note; 0 for default>,`
		`"String:=", <string note 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	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.

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.

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

			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"])`
	`oModule.AddTraceCharacteristics("Rept2DRectFreq", "distortion", ["0"], ["Specified", "19.5GHz", "20.4GHz"])`

AddTraces

Creates a new trace and adds it to the 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.

<table>
<tr><td>UI Access</td><td>Modify Report > Add Trace.</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</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution as listed in the Modify Report dialog box.</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.
["X Component:=", <VariableName>,
"Y Component:=", <VariableName> | <ReportQuantityArray>]
<ReportQuantityArray> ex. ["dB[S[Port1, Port1]]"]</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 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.

UI Access	Right-click on a report, select **Report Templates** > **Apply Settings**.		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of the report to apply settings to.
	<*TemplateFile*>	String	Template file name with path.
	<*PropertyType*>	String	Property types to apply. `("Graphical" | "Data" | "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	ApplyReportTemplate(<*ReportName*>, <*TemplateFile*>, <*PropertyType*>)
Python Example	`oModule.ApplyReportTemplate("Variables Plot 1", "C:/MyTemplate.rpt", "Graphical")`

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	Name | Type | Description <*PlotName*> | String | Name of the plot
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	Name | Type | Description <*ReportsToClone*> | Array | Array of report names to clone. <*SoluNameToUse*> | String | Name of the dataset solution.
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

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 Report
	<*TracesArray*>	Array	Trace definitions to copy
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</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.

			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 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>

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

			["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.		

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]`
	`],`

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

```
        [
                "Time:="       , ["All"],
                "aaa:="        , ["Nominal"]
        ],
        [
                "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",
```


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", "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"]
        ],
        [
                "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"]
        ],
        [
```


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:=", "GasPressure",
                "Y Component:=", ["Power"]
        ])
```

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",
```


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"
                                "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))"
                        ],
                        [
```


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

```
rs",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "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
    ],
```


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")
```

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

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	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*>,

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

	<*SolutionName*>, <*SimValueCtxt*>, <*CategoryName*>, <*PointSet*>, <*CommonComponentsOfTraces*>, <*ExtTraceInfo*>)
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")`

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

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.
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"])`

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

DeleteTraceCharacteristics

Deletes a trace characteristics field from a report

UI Access	N/A
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 delete from.</td></tr><tr><td><*TraceCharsNames*></td><td>Array</td><td>Array of trace characteristics to delete.</td></tr></table>
Return Value	None.

Python Syntax	DeleteTraceCharacteristics(<*ReportName*>, <*TraceCharsNames*>)
Python Example	`oModule.DeleteTraceCharacteristics("Variables Plot 1", ["lowercutoff", "gain-crossover"])`

DeleteTraces

Deletes an existing traces or traces.

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.</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.

			["<ReportName>:=", <TracesArray>, <TracesArray>,...]
	<*ReportName*>	String	Name of Report
	<*TracesArray*>	Array	Contains the traces to delete within a report [<Trace>, <Trace>, ...]
	<*Trace*>	String	A specific trace that the user wishes to delete
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	Name	Type	Description
	<*DisplayType*>	String	Specified display type to check.
Return Value	Integer • **1** - trace characteristics is supported. • **0** - trace characteristics not supported.		

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	DoesSupportTraceCharacteristics(<*DisplayType*>)
Python Example	`oModule.DoesSupportTraceCharacteristics("Rectangular Plot")`

DumpAllReportsData

Dumps all reports data to an Ansoft report data file.

UI Access	N/A
Parameters	Name | Type | Description <*FileName*> | String | File name with path.
Return Value	None.

Python Syntax	DumpAllReportsData(<*FileName*>)
Python Example	`oModule.DumpAllReportsData("C:/ReportsData.rdat")`

EditCartesianXMarker

Edits an XMarker value.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of report.

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

	<*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)`

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,`

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

	"dB() : Setup1 : Sweep1")

EditMarker

Edits a marker on a report.

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 report.</td></tr><tr><td><*MarkerName*></td><td>String</td><td>Marker name, including any trailing number.</td></tr><tr><td><*CurveName*></td><td>String</td><td>Full trace name.</td></tr><tr><td><*PrimarySweepValue*></td><td>String</td><td>Primary sweep value, including unit.</td></tr></table>
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.

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><*ReportName*></td><td>String</td><td>Name of specified report.</td></tr><tr><td><*ExportFileName*></td><td>String</td><td>File name to export to.</td></tr></table>
Return Value	N/A

Python Syntax	ExportEyeMaskViolation(<*ReportName*>, <*ExportFileName*>)
Python Example	`oModule.ExportEyeMaskViolation("Variables Plot 1", "C:/eyemask1.csv")`

ExportImageToFile [Reporter]

Exports a report image in a specified format.This command is fully supports -ng (non-graphical) mode.

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 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</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, False or Default. Default is True.
Return Value	None		

Python Syntax	ExportImageToFile(<*ReportName*>, <*FileName*>, <*Width*>, <*Height*>)

Python Example	#### Script Recorded for Saving 2D
	`oDesktop.OpenProject("E:/Ansoft/Tee2.aedt")`
	`oProject = oDesktop.SetActiveProject("Tee2")`
	`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"`

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 1", "E:/Ansoft/S Parameter Plot 1.png",
1920, 1080,
  [
    "NAME:SaveImageParams",
    "ShowLegend:=" , "True",
    "ShowMarkerTable:=" , "True"
    "ShowDeltaMarkerTable:=" , "False"
  ])
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"
```


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

```
    "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")
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",
```


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

```
    "AutoFit:=" , "False",
    "Orientation:=" , "",
    "ShowOrientationGadget:=", "Default"
    "ShowLegend:=" , "True"
  ])
oModule.ExportImageToFile("S Parameter Plot 6", "E:/Ansoft/S Parameter Plot 10.png",
1335, 466,
  [
    "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:

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

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

ExportModelImageToFile supports export overlay of polar plot 3D with existing transformation (scaling, rotation and translation) in -ng (non-graphical) mode.

UI Access

Modeler > **Export**.

Parameters

Name	Type	Description
<*path*>	String	Full file path including extension.
<*width*>	Integer	Width in pixels (use 0 for default).
<*height*>	Integer	Height in pixels (use 0 for default).

<*Parameters*>	Array	Structured array.
		`["NAME:SaveImageParams",`
		`"ShowAxis:=" , <string containing boolean>,`

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

```
"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:=" , "",
```

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.

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

			"Orientation:=" , <string> "ShowOrientationGadget:=" , <False>]
Return Value	None.		

Python Syntax	ExportModelImageToFile(<*path*> <*width*> <*height*> <*Parameters*>)
Python Example	oEditor.ExportModelImageToFile ("D:/Image.png", 1920, 1080, ["NAME:SaveImageParams", "ShowAxis:=" , "True", "ShowGrid:=" , "True", "ShowRuler:=" , "True", "ShowRegion:=" , "Default", "Selections:=" , "", "FieldPlotSelections:=" , "", "FitToSelections:=" , "", "FitToFieldPlotSelections:=", "", "AutoFit:=" , "True", "Orientation:=" , ""])

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

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.

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>`

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>

Path to file.

Python Syntax	ExportPlot3DToFile(<*name*> <*Path*>)

Python Example	`oModule = oDesign.GetModule("ReportSetup")`
	`oModule.UpdateReports(["rE Plot 1"])`
	`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.

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 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		

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.		

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	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.
<*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")`

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><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
.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>

Python Syntax	ExportToFile(<*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.ExportToFile("rETotal", "C:/Users/Documents/rETotal.csv")` `oModule.ExportToFile("S Parameter Table 1", "D:/Users/Documents/cftt.csv", False, " kV, MHz, yd ", True)`

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of the report</td></tr><tr><td><*FileName*></td><td>String</td><td>Path and File Name Supported formats<table><tr><td>.txt</td><td>Post processor format file</td></tr><tr><td>.csv</td><td>Comma-delimited data file</td></tr><tr><td>.tab</td><td>Tab-separated file</td></tr><tr><td>.dat</td><td>Ansys plot data file</td></tr><tr><td>.rdat</td><td>Ansys report data file</td></tr></table></td></tr></table>
Return Value	None

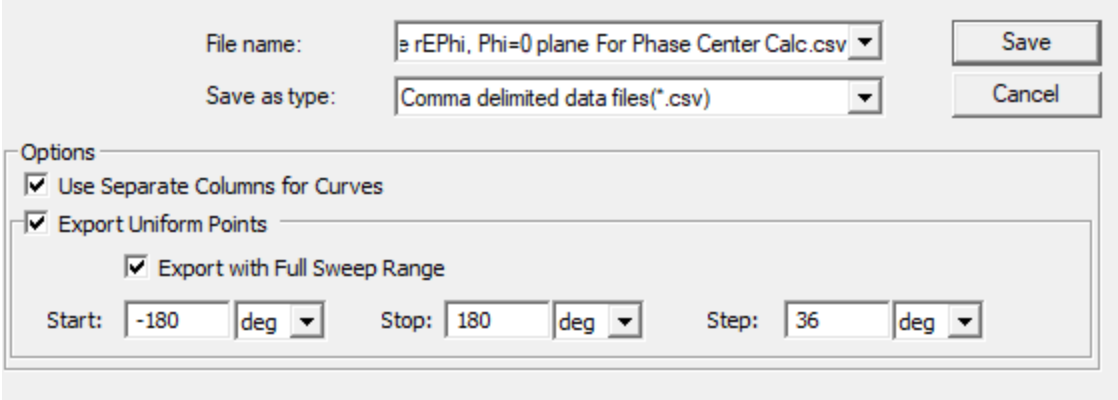
Python Syntax	ExportToFile (<*ReportName*>, <*FileName*>)
Python Example	`oModule.ExportToFile(`

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

	"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**.
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

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

			.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.

UI Access	NA		
Parameters	Name	Type	Description
	<*ReportType*>	String	Report type name.
	<*DisplayType*>	String	Name of display type.
	<*SolutionName*>	String	Name of solution.
	<*SimValueCtxt*>	Array	A context name or array of strings that encode the contexts.
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	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	

Name	Type	Description
<*ReportType*>	String	Report type name.
<*DisplayType*>	String	Display type name.
<*SolutionName*>	String	Name of solution.
<*SimValueCtxt*>	Array	A context name, or array of strings that encoded the contexts(I).
<*CategoryName*>	String	A category name as input parameter. a category name returned in GetAllCategories() or "Variables", or "Output Variables"

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")`

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

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()`

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

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	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()`

GetAvailableSolutions

Gets all available solutions in report type 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></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	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.

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

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

	<*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.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing child object types.

Python Syntax	GetChildTypes ()
Python Example	`oModule.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

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")`

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.

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	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		

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

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><*TraceName*></td><td>String</td><td>Name of specified source trace.</td></tr><tr><td><*QtyName*></td><td>String</td><td>Name of specified quantity value.</td></tr></table>
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.

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 the source solution. If empty, refer to current solution.</td></tr></table>
Return Value	Array of strings containing trace names.

Python Syntax	GetDynLinkTraces(<*SoluName*>)
Python Example	`oModule.GetDynLinkTraces("")`

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

GetDynLinkVariableValues

Gets the values of a variable from a dynamic linked trace.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*TraceName*></td><td>String</td><td>Name of specified dynamic linked trace, with its report name.</td></tr><tr><td><*VarName*></td><td>String</td><td>Name of specified variable.</td></tr></table>
Return Value	Array of strings containing variable values.

Python Syntax	GetDynLinkVariableValues(<*TraceName*>, <*VarName*>)
Python Example	`oModule.GetDynLinkVariableValues("Plot 1:Trace", "Var1")`

GetName

Returns the design name of the active design, in that order separated by a semicolon.

UI Access	N/A
Parameters	None.
Return Value	String indicating the name of the active design.

Python Syntax	GetName()
Python Example	design_name = oDesign.GetName()

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()`

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:

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 `Comp:Inst@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.

GetPropNames [Reporter]

Report setup module does not have its own property, this function always returns empty array.

UI Access	N/A
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*>)

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.GetPropValue("S Parameter Plot 1/Display Type")`

GetQtyExpressionsForSourceTrace

Gets the quantity expressions from a specified source trace.

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.

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	GetReportTraceNames(<*PlotName*>)
Python Example	`oModule.GetReportTraceNames("SParameter Plot 1")`

GetReportSummaryForRegressionTesting

Gets report summary from a dumped report file.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of a specified dumped report.
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	Name | Type | Description <*ReportType*> | String | Report type name.

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

	<DisplayType>	String	Display type name.
	<SolutionName>	String	Name of solution.
Return Value	Array of text strings		

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	Name	Type	Description
	<reportTypeArg>	String	Report type name as input parameter.
	<solutionNameArg>	String	Solution name as input parameter.
	<simValueCtxtArg>	Structured Array	Same as ContextArray values created in the relevant CreateReport script.
	<familiesArg>	Array of Strings	Same as FamiliesArray values created in the relevant CreateReport script.
	<expressionArg>	String or Array of Strings	Text string or array of text strings; valid expression, may validate it as the data-table Y-component.
Return Value	ARRAY of ISolutionDataResultComInterface objects, containing: • GetSweepNames()		

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

	• GetSweepUnits() • GetSweepValues() • IsPerQuantityPrimarySweep() • GetPerQuantityPrimarySweepValues() • 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, expres-

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

	sionArg)

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))'])`

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)`

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

GetDesignVariableNames

Returns array of strings containing design variable names.

> **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.

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	<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

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><varName></td><td>String</td><td>Can be returned from GetDesignVariableNames()</td></tr><tr><td><siValue></td><td>Boolean</td><td>True to return SI-value; False to return by GetDesignVariableUnits()</td></tr></table>
Return Value	Double value

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.

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	GetDesignVariationKey()
Python Example	`oModule.GetDesignVariationKey()`

GetImagDataValues

Returns array of imaginary data 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
Parameters	Name | Type | Description <expressionString> | String | Can be returned from `GetDataExpressions()` <siValue> | Integer | 1 to return SI-value; 0 to return with units returned in GetSweepUnits().
Return Value	Array of doubles

Python Syntax	GetImagDataValues(<expressionString>,<siValue>)
Python Example	`imaginaryvalues = oModule.GetImagDataValues('expression',1)`

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

GetPerQuantityPrimarySweepValues

Returns per quantity 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
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.

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	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.

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	N/A
Return Value	Array of text strings

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 | Type | Description <sweepName> | String | Primary sweep name
Return Value	Text string containing units

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	GetSweepUnits(<sweepName>)
Python Example	`sweepunits = oModule.GetSweepUnits('Sweep 1')`

GetSweepValues

Returns 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
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)`

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

IsDataComplex

Returns whether an expression is complex.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	NA
Parameters	Name: <expressionString> — Type: String — Description: Can be returned from `GetDataExpressions()`.
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.

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	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

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	N/A

Python Syntax	ReleaseData()
Python Example	`oModule.ReleaseData()`

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")`

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

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of the Report</td></tr><tr><td><*FileName*></td><td>String</td><td>Path and File Name<table><tr><td>.csv</td><td>Comma-delimited data file</td></tr><tr><td>.tab</td><td>Tab-separated file</td></tr><tr><td>.dat</td><td>Ansys plot data file</td></tr></table></td></tr></table>
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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of specified report.</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.

	<*FileName*>	String	Name of specified data file. File extenstion "rdat" is expected.
Return Value	None.		

Python Syntax	ImportReportDataIntoReport(<*ReportName*>, <*FileName*>)
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")`

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

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
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><*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.

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	OpenWindowForAllReports()
Python Example	`oModule.OpenWindowForAllReports()`

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**

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 specified report.</td></tr><tr><td><*PropTypeToApply*></td><td>String</td><td>Property type to paste. "Graphical", "Data", or "All".</td></tr></table>
Return Value	None.

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()`

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

PasteReportsWithLegacyNames

Pastes copied reports to results in the current project with legacy name definitions.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	PasteReportsWithLegacyNames ()
Python Example	`oModule.PasteReportsWithLegacyNames()`

PasteTraces

Pastes copied traces to a named plot.

UI Access	Paste
Parameters	Name: *<ReportName>*; Type: String; Description: Name of plot
Return Value	None

Python Syntax	PasteTraces (<*ReportName*>)

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.PasteTraces ("XY Plot1")`

PasteTracesWithLegacyNames

Pastes copied traces to a named plot using legacy name definitions.

UI Access	N/A
Parameters	Name | Type | Description *<ReportName>* | String | Name of plot
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	Name | Type | Description *<OldReportName>* | String | Old Report Name *<NewReportName>* | String | New Report 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	RenameReport (<*OldReportName*>, <*NewReportName*>)
Python Example	`oModule.RenameReport("XY Plot1", "Reflection")`

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")`

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

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.

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*>")

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.SavePlotSettingsAsDefault("XY Plot1")`

SetLinkOutputTraces

Specifies dynamic link output traces from the current design.

UI Access	Right-click the **Results**, select **Link Output...**		
Parameters	Name	Type	Description

	<*TraceArray*>	Array	Array of traces to set.
			`["<ReportName>:=", <array of trace names>,`
			`"<ReportName>:=", <array of trace names>,...]`

Return Value	None.		

Python Syntax	SetLinkOutputTraces(<*TraceArray*>)

Python Example	`oModule.SetLinkOutputTraces(`
	`[`
	`  "Plot 1:=", ["Trace1"],`
	`  "Plot 2:=", ["Trace1"]`
	`])`

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

SetPropValue [Report Setup]

Sets the property value for report module child object.

UI Access	Select **Edit Properties** on Report objects.
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 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.

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	<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><*GroupName*></td><td>String</td><td>Stack group name.</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	UnGroupPlotCurvesInGroup(<*ReportName*>, <*GroupName*>)
Python Example	`oModule.UngroupPlotCurvesInGroup("S Parameter Plot 3", "Stack 1")`

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.

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
	<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"])`

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.

UI Access	In **Report** dialog, click **Apply Traces** button		
Parameters	Name	Type	Description
	<ReportName>	String	Name of Report.
	<TraceNames>	Array	Array of strings containing trace names.
	<SolutionName>	String	Name of the solution.

	<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 Spheren", "Spheren",`

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

			"Polylinen"
	<*FamiliesArray*>	Array	Contains sweep definitions for the report. ["<VariableName>:= ", <ValueArray>] <ValueArray> ["All"] or ["Value1", "Value2", ..."Valuen"] 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))"]
	<*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,

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

	"100","TE",False,"100s","TH",False,"40", "TS",False,"0ns","UF",False, "0","WT",False,"0","WW",False,"100"]], ["Spectrum:=", ["All"]], ["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	

Name	Type	Description
<*ReportName*>	String	Name of Report.
<*TraceNames*>	Array	Array of strings containing trace names.
<*SolutionName*>	String	Name of the solution as listed in the **Modify Report** dialog box. For example: "Setup1 : Last Adaptive"
<*ContextArray*>	Array	Context for which the expression is being evaluated. This can be an empty

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

			string if there is no context. ex. "Sweep" or "Time"
	<*PointSet*>	Array	Point set for the selected traces, for example, X and Y values for the plot.
Return Value	None		

Python Syntax	UpdateTracesContextAndSweeps(<*ReportName*>, <*TraceNames*>, <*SolutionName*>, <*ContextArray*>, <*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.

11 - Boundary and Excitation Module Script Commands

Boundary condition commands should be executed by the "BoundarySetup" module.

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

Conventions Used in this Chapter

`<BoundName>`

Type: string.

Name of a boundary.

`<AssignmentObjects>`

Type: Array of strings.

An array of object names.

`<AssignmentFaces>`

Type: Array of integers.

An array of face IDs. The ID of a face can be determined through the user interface using the **3D Modeler** > **Measure** > **Area** command. The face ID is given in the **Measure Information** dialog box.

`<LineEndPoint>`

```
  [<double>, <double>, <double>]
```

The topics for this section include:

General Commands Recognized by the Boundary/Excitations Module

Script Commands for Creating and Modifying Boundaries

Script Commands for Creating and Modifying PMLs

General Commands Recognized by the Boundary Conditions Module

AddAssignmentToBoundary

DeleteAllBoundaries

DeleteBoundaries

GetBoundaryAssignment

GetBoundaries

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

GetBoundariesOfType

GetDefaultBaseName

GetNumBoundaries

GetNumBoundariesOfType

ReassignBoundary

RemoveAssignmentFromBoundary

RenameBoundary

ReprioritizeBoundaries

SetDefaultBaseName

AddAssignmentTo

Adds a new geometry assignment to a boundary. To use this command, first call the BoundarySetup module, and then call AddAssignmentTo.

> **Note:** This command replaces the AddAssignmentToBoundary command of previous releases.

UI Access	N/A

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Assignment*></td><td>Array</td><td>Structured array.
	`["Name:<BoundName>",`
	`"Objects:=", <AssignmentObjects>,`
	`"Faces:=", <AssignmentFaces>]`</td></tr></table>

Return Value	None.

Python Syntax	AddAssignmentTo(<*Assignment*>)

Python Example	`oModule = oDesign.GetModule("BoundarySetup")`
	`oModule.AddAssignmentTo(`
	`[`
	`"NAME:Impedance1_tank",`
	`"Objects:=" , ["LVA"]`
	`])`

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

DeleteAllBoundaries

Deletes all boundaries.

UI Access	[product] > **Boundaries** > **Delete All**
Parameters	None.
Return Value	None.

Python Syntax	DeleteAllBoundaries()
Python Example	`oModule.DeleteAllBoundaries()`

Delete Boundaries

To delete specified boundaries or excitations, first call the BoundarySetup module, and then call the Delete command.

> **Note:** The Delete command replaces the DeleteBoundaries of previous releases.

UI Access	**Delete** command in the **List** dialog box. Click **[product]** > **List** to open the **List** dialog box.
Parameters	Name | Type | Description *<NameArray>* | Array | Array of boundary condition names.
Return Value	None.

Python Syntax	Delete(*<NameArray>*)
Python Example	`oModule = oDesign.GetModule("BoundarySetup")` `oModule.Delete(["PerfE1", "WavePort1"])`

Edit for BoundarySetup Commands

For existing boundaries and excitations, the Edit command can be used to change properties. To use this command, first call the BoundarySetup module, and then with the Edit command,

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 boundary or excitation and then the properties of that particular boundary or excitation. For example, to change the properties of a current excitation (Maxwell):

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("Current2",
  [
    "NAME:Current2",
    "Phase:=" , "0deg",
    "Current:=" , "5A",
    "IsSolid:=" , False,
    "Point out of terminal:=", True
  ])
```

Important:

The Edit command replaces the EditBoundaryName commands of previous releases.

Note:

The Edit command cannot be used to change the entity that the boundary or excitation is assigned to. To change the entity, use one of the following commands:

- AddAssignmentTo
- RemoveAssignmentFrom
- Reassign

UI Access	Right click on the boundary or excitation in the Project Manager tree, and select **Properties**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*Name*></td><td>String</td><td>Name of the boundary or excitation to be edited.</td></tr><tr><td><*Properties*></td><td>Array</td><td>Structured array. The required contents of this array is determined by the type of boundary or excitation specified in the Name parameter. See the corresponding AssignBoundaryName topic for a list of properties that can be edited.</td></tr></table>
Return Value	None

Python Syntax	Edit(<*Name*>, <*Properties*>)

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

HFSS:

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("FiniteCond1",
["NAME:FiniteCond1",
  "Roughness:=", "2um",
  "UseCoating:=", false
])
```

Maxwell:

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("VectorPotential1",
  [
    "NAME:VectorPotential1",
    "Phase:="               , "30deg",
    "Value:="               , "2"
  ])
```

IcepackFEA:

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("Contact1",
  [
    "NAME:Contact_Top",
    "Resistance Type:=" , "Thickness"
    "Thickness:="       , "2.5mil",
    "Material:="        , "polyehtylene"
  ])
```

Q3D:

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("ContactResistance1"
  [
    "NAME:ContactResistance1",
    "Contact Resistance Type:=", "Thin Layer",
    "Conductivity:="        , "58000000S_per_m",
    "Thickness:="           , "2mm"
  ])
```

Icepak:

```
oModule = oDesign.GetModule("BoundarySetup")
oModule.Edit("Opening1",
  [
```


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

```
    "NAME:Opening1",
    "Temperature:="         , "AmbientTemp",
    "External Rad. Temperature:=", "AmbientRadTemp",
    "Inlet Type:="          , "Velocity",
    "Static Pressure:="     , "AmbientPressure",
    "X Velocity:="          , "2m_per_sec"
  ])
```

GetBoundaries

Gets boundary names in the current design.

UI Access	N/A
Parameters	None.
Return Value	Array of boundary names.

Python Syntax	GetBoundaries()
Python Example	`oModule.GetBoundaries()`

GetBoundariesOfType

Gets boundary names of the given type.

UI Access	N/A
Parameters	Name | Type | Description <*BoundaryType*> | String | Name of boundary type.
Return Value	Array of boundary names of the given type.

Python Syntax	GetBoundariesOfType(<*BoundaryType*>)
Python Example	`oModule.GetBoundariesOfType("PerfectE")`

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

GetAssignment for Boundaries or Excitations

Gets a list of face IDs associated with the given boundary or excitation assignment. First call the BoundarySetup module, and then call Get Assignment.

> **Note:** GetAssignment replaces the GetBoundaryAssignment command of previous releases.

UI Access	N/A
Parameters	Name: <*BoundaryName*> — Type: String — Description: Name of the specified boundary or excitation.
Return Value	Array of face IDs or object IDs.

Python Syntax	GetAssignment(<*BoundaryName*>)

Python Example	`oModule = oDesign.GetModule("BoundarySetup")`
	`oModule.GetAssignment("VectorPotential1")`
	Returns
	`['4535', '4536', '4537', '4538']`

GetDefaultBaseName

Gets the default base name for boundaries for a project.

UI Access	N/A
Parameters	Name: <*BoundaryType*> — Type: String — Description: Name of legal boundary type.
Return Value	String of boundary default base name.

Python Syntax	GetDefaultBaseName(<*BoundaryType*>)
Python Example	`oModule.GetDefaultBaseName("Radiation")`

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

GetNumBoundaries

Gets the number of boundaries in a design.

UI Access	N/A
Parameters	None.
Return Value	Integer number of boundaries.

Python Syntax	GetNumBoundaries()
Python Example	`oModule.GetNumBoundaries()`

GetNumBoundariesOfType

Gets the number of boundaries of the given type.

UI Access	N/A
Parameters	Name | Type | Description <*BoundaryType*> | String | Specified boundary type.
Return Value	Integer number of boundaries.

Python Syntax	GetNumBoundariesOfType(<*BoundaryType*>)
Python Example	`oModule.GetNumBoundariesOfType("PerfectE")`

Reassign a Boundary

Use the Reassign command to specify a new geometry assignment for a boundary. First call the BoundarySetup module, and then call the Reassign command.

> **Note:** This command replaces the ReassignBoundary command of previous releases.

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 **Boundaries** > **Reassign** or **Excitations** > **Reassign**

Parameters	Name / Type / Description
	<AssignmentArray> / Array / Structured array.
	`["Name:<BoundName>",`
	`"Objects:=", <AssignmentObjects>,`
	`"Faces:=", <AssignmentFaces>]`

Return Value	None.

Python Syntax	ReassignBoundary(*<AssignmentArray>*)

Python Example	`oModule = oDesign.GetModule("BoundarySetup")`
	`oModule.Reassign(`
	`[`
	`"NAME:PerfH12",`
	`"Faces:=", [17]`
	`])`

RemoveAssignmentFrom

Removes a geometry assignment from a boundary. To use this command, first call the BoundarySetup module, and then call RemoveAssignmentFrom.

> **Note:** This command replaces the RemoveAssignmentFromBoundary command of previous releases.

UI Access	Right-click on a boundary or an excitation, select **Remove from Assignment**.

Parameters	Name / Type / Description
	<AssignmentArray> / Array / Structured array.
	`["Name:<BoundName>",`
	`"Objects:=", <AssignmentObjects>,`
	`"Faces:=", <Assign-`

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

	mentFaces>]
Return Value	None

Python Syntax	RemoveAssignmentFrom(<*AssignmentArray*>)

Python Example	`oModule.RemoveAssignmentFrom(`
	`[`
	`"NAME:Rad1",`
	`"Faces:=", [11]`
	`])`

RenameBoundary

This command has been deprecated. It has been replaced by Rename.

ReprioritizeBoundaries

Specifies the order in which the boundaries and excitations are recognized by the solver. The first boundary in the list has the highest priority.

> **Important:** This command is only valid if all defined boundaries and excitations appear in the list. All ports must be listed before any other boundary type.

UI Access	[**product**] > **Boundaries** > **Reprioritize**		
Parameters	Name	Type	Description
	<*NewOrderArray*>	Array	Structured array. `["NAME:NewOrder",` <BoundName>, <BoundName>, ...]
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	ReprioritizeBoundaries(<*NewOrderArray*>)

Python Example	`oModule.RepriorizitizeBoundaries(`
	`  ["NAME:NewOrder",`
	`  "Imped1",`
	`  "PerfE1",`
	`  "PerfH1"`
	`  ])`

SetDefaultBaseName

Sets the default base name for boundaries for a project.

UI Access	N/A		
Parameters	Name	Type	Description
	<*BoundaryType*>	String	Name of legal boundary type.
	<*DefaultName*>	String	Default name for boundaries of specified type.
Return Value	None.		

Python Syntax	SetDefaultBaseName(<*BoundaryType*>, <*DefaultName*>)
Python Example	`oModule.SetDefaultBaseName("Radiation", "RadBnd")`

Icepak Boundary Condition Commands

Icepak thermal boundary condition commands should be executed by the oModule object. For example, use `oModule.AssignBlockBoundary` to assign a block thermal boundary condition.

See the following sections to view the parameters for each boundary condition.

AssignBlockBoundary

AssignGrilleBoundary

AssignNetworkBoundary

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

AssignOpeningBoundary

AssignRecircBoundary

AssignAdiabaticPlateBoundary

AssignConductingPlateBoundary

AssignResistanceBoundary

AssignSourceBoundary

AssignStationaryWallBoundary

AssignSymmetryWallBoundary

AssignEMLoss

AssignBlowerBoundary

AssignInitialTemperature

AssignBlockBoundary

Creates a block thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Block**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Objects>	list	Objects included in the boundary condition assignment
	<Block Type>	string	Solid, Hollow, or Fluid
	<Use External Conditions>	bool	"True" to enable external thermal conditions or "False" to disable them
	<Heat Transfer Coefficient>	int	Heat transfer coefficient value and unit
	<Temperature>	int	Block temperature value and unit
	<Use Total Power>	bool	True or False
	<Total Power>	int	Block power value and unit
	<Power Density>	string	Block power density value and unit
	<NAME>	string	Total Power Variation Data
	<Variation Type>	string	Temp Dep, Transient, or Joule Heating
	<Variation Function>	string	*Temp Dep*: Piecewise Linear; *Transient*: Linear, Power Law, Exponential

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

			Sinusoidal, or Piecewise Linear; *Joule Heating*: None
	<*Variation Value*>	string	Temperature dependant or Transient variation function values.
	<*Thermal Condition*>	string	Hollow blocks only: Fixed Heat, Fixed Temperature, or Internal Conditions
	<*Use Total Power*>	bool	Hollow blocks only: "True" to enable total power or "False" to disable it
	<*Use Heat Flux*>	bool	Hollow blocks only: "True" to enable heat flux or "False" to disable it
	<*Use Laminar Flow*>	bool	Fluid blocks only: "True" to enabled laminar flow or 'False' to disable it
	<*Use MRF*>	bool	Fluid cylinders only: "True" to enabled moving reference frame or "False" to disable it
	<*MRF*>	int	Fluid cylinders only: Moving reference frame value and unit
	<*X Fixed Velocity*>	string	X direction fixed velocity value and unit
	<*Y Fixed Velocity*>	string	Y direction fixed velocity value and unit
	<*Z Fixed Velocity*>	string	Z direction fixed velocity value and unit
Return Value	None		

Python Syntax	AssignBlockBoundary (<*NAME*>, <*Objects*>, <*Block Type*>, <*Use External Conditions*>, <*Heat Transfer Coefficient*>, <*Temperature*>, <*Total Power*>)

Python Example	`oModule.AssignBlockBoundary(`
	`[`
	`"NAME:Block1",`
	`"Objects:=" , ["Box1"],`
	`"Block Type:=" , "Solid",`
	`"Use External Conditions:=", True,`
	`"Heat Transfer Coefficient:=", "1w_per_m2kel",`
	`"Temperature:=" , "AmbientTemp",`

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

```
                "Total Power:="         , "0W"

                 [
                        "NAME:Total Power Variation Data",
                        "Variation Type:="      , "Temp Dep",
                        "Variation Function:="  , "Piecewise Linear",
                        "Variation Value:="     , "[\"1W\", \"1\"]"
                ]
        ])

        ])
```

AssignGrilleBoundary

Creates a grille thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Grille**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Faces>	list	Faces included in the boundary condition assignment
	<Pressure Loss Type>	string	Coeff or Curve
	<Free Area Ratio>	string	Coeff only: greater than 0 and less than or equal to 1
	<Resistance Type>	string	Coeff only: Perforated Thin Vent, Circular Metal Wire Screen, or Two-Plane Screen Cyl. Bars
	<External Rad. Temperature>	int	External radiation temperature value and unit
	<External Total Pressure>	int	External total pressure value and unit
	<NAME>	string	Curve only: dataset name
	<X>	list	Curve only: loss curve dataset x values

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

	<*Y*>	list	Curve only: loss curve dataset y values
Return Value	None		

Python Syntax	AssignGrilleBoundary (<*NAME*>, <*Faces*>, <*Pressure Loss Type*>, <*Free Area Ratio*>, <*Resistance Type*>, <*External Rad. Temperature*>, <*External Total Pressure*>, <*NAME*>, <*X*>, <*Y*>)

Python Example	`oModule.AssignGrilleBoundary(`
	`  [`
	`  "NAME:Grille1",`
	`  "Faces:=" , [1],`
	`  "Pressure Loss Type:=" , "Coeff",`
	`  "Free Area Ratio:=" , "0.8",`
	`  "Resistance Type:=" , "Perforated Thin Vent",`
	`  "External Rad. Temperature:=", "AmbientTemp",`
	`  "External Total Pressure:=", "AmbientPressure",`
	`  [`
	`  "NAME:DimUnits",`
	`  "",`
	`  ""`
	`  ],`
	`  "X:=" , ["0","1","2"],`
	`  "Y:=" , ["0","1","2"]`
	`  ])`

AssignNetworkBoundary

Creates a network thermal boundary condition in an Icepak design. See the Icepak Online Help for example scripts.

UI Access	**Icepak** > **Thermal** > **Assign** > **Network**
Parameters	

Name	Type	Description
<*NAME*>	string	Boundary condition/component name
<*Faces*>	list	Faces included in the boundary condition assignment
<*Face*>	string	Face node (Face number and resistance choice)
<*Internal*>	string	Internal node (power and unit, mass and unit, specific heat and unit)
<*Link1*>	string	Link (linked nodes, Rlink type, and thermal resistance or heat transfer

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

			coefficient)
	<SchCompInst>	list	Schematic component instance (schematic coordinates)
	<NetName>	string	Wiredata net name
	<ID>	int	Node ID
	<WireSeg>	list	Coordinates of wire segment
	<Color>	int	Numerical color value
Return Value	None		

AssignOpeningBoundary

Creates a free opening thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Opening** > **Free**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Faces>	list	Faces included in the boundary condition assignment
	<Temperature>	string	Opening temperature value and unit
	<External Rad. Temperature>	string	External radiation temperature value and unit
	<Inlet Type>	string	Pressure, Velocity, or Mass Flow
	<Total Pressure>	string	Pressure only: Total pressure value and unit
	<Static Pressure>	string	Velocity and Mass Flow only: Static pressure value and unit
	<Mass Flow Rate>	string	Mass Flow only: Mass flow rate and unit
	<Mass Flow Direction>	string	Normal to Boundary or Specified
	<Mass Flow Condition>	string	In Flow or Out Flow
	<X Velocity>	string	X direction velocity and unit
	<Y Velocity>	string	Y direction velocity and unit
	<Z Velocity>	string	Z direction velocity and unit
	<No Reverse Flow>	string	True or False
Return Value	None		

Python Syntax	AssignOpeningBoundary (*<NAME>*, *<Faces>*, *<Temperature>*, *<External Rad. Temperature>*, *<Inlet Type>*, *<Total Pressure>*)

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.AssignOpeningBoundary(`
	`  [`
	`  "NAME:Opening1",`
	`  "Faces:=" , [12],`
	`  "Temperature:=" , "AmbientTemp",`
	`  "External Rad. Temperature:=", "AmbientRadTemp",`
	`  "Inlet Type:=" , "Pressure",`
	`  "Total Pressure:=" , "AmbientPressure"`
	`  ])`

AssignRecircBoundary

Creates a recirculating opening thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Opening** > **Recirculating**
Parameters	

Name	Type	Description
<*NAME*>	string	Boundary condition name
<*Faces*>	list	Faces included in the boundary condition assignment
<*Thermal Condition*>	string	Temperature, Heat Input, or Conductance
<*Temperature Change* (Temperature only)>	string	Temperature value and unit
<*Heat Flow* (Heat Input only)>	string	Heat flow value and unit
<*Conductance* (Conductance only)>	string	Conductance value and unit
<*Inlet Type*>	string	Mass Flow, Mass Flux, or Volume Flow
<*Mass Flow Rate* (Mass Flow only)>	string	Mass flow rate value and unit
<*Mass Flux Rate* (Mass Flux only)>	string	Mass flux value and unit
<*Volume Flow Rate* (Volume Flow only)>	string	Volume flow rate value and unit
<*Supply Flow Direction*>	string	Please type in description manually
<*X, Y, and Z* (Supply Flow Direction: Specified only)>	int	Direction vector values
<*ExtractFace*>	list	Extract opening face

Return Value	None
Python Syn-	AssignRecircBoundary (<*NAME*>, <*Faces*>, <*Thermal Condition*>, <*Tem-*

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

tax	*perature Change*>, <*Inlet Type*>, <*Mass Flow Rate*>, <*Supply Flow Direction*>, <*ExtractFace*>)

Python Example	`oModule.AssignRecircBoundary(`
	`  [`
	`  "NAME:RecircOpening1",`
	`  "Faces:=" , [40,35],`
	`  "Thermal Condition:=" , "Temperature",`
	`  "Temperature Change:=" , "0cel",`
	`  "Inlet Type:=" , "Mass Flow",`
	`  "Mass Flow Rate:=" , "0kg_per_s",`
	`  "Supply Flow Direction:=", "Normal",`
	`  "ExtractFace:=" , [40]`
	`  ])`

AssignAdiabaticPlateBoundary

Creates an adiabatic plate thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Plate** > **Adiabatic**
Parameters	

Name	Type	Description
<*NAME*>	string	Boundary condition name
<*Faces*>	list	Faces included in the boundary condition assignment
<*NAME*>	string	LowSide
<*Radiate*>	string	LowSide: "True" to enable low side for radiation or "False" to disable it
<*RadiateTo*>	string	LowSide: AllObjects or RefTemperature
<*Ref. Temperature*>	string	LowSide: Reference temperature and unit
<*View Factor*>	string	LowSide: View factor
<*Surface Material*>	string	LowSide: Surface material
<*NAME*>	string	HighSide
<*Radiate*>	string	HighSide: "True" to enable high side for radiation or "False" to disable it
<*RadiateTo - High*>	string	HighSide: AllObjects or RefTemperature - High

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

	<Ref. Temperature - High>	string	HighSide: Reference temperature and unit
	<View Factor - High>	string	HighSide: View factor
	<Surface Material - High>	string	HighSide: Surface material
Return Value	None		

Python Syntax	AssignAdiabaticPlateBoundary (*<NAME>*, *<Faces>*, *<NAME>*, *<Radiate>*, *<NAME>*, *<Radiate>*)

Python Example	`oModule.AssignAdiabaticPlateBoundary(`
	`  [`
	`  "NAME:AdiabaticPlate1",`
	`  "Faces:=" , [7],`
	`  [`
	`  "NAME:LowSide",`
	`  "Radiate:=" , True,`
	`  "RadiateTo:=" , "AllObjects",`
	`  "Surface Material:=" , "Steel-oxidised-surfac`
	`  ],`
	`  [`
	`  "NAME:HighSide",`
	`  "Radiate:=" , True,`
	`  "RadiateTo - High:=" , "RefTemperature - High`
	`  "Ref. Temperature - High:=", "0",`
	`  "View Factor - High:=" , "0",`
	`  "Surface Material - High:=", "Steel-oxidised-sur`
	`  ]`
	`  ])`

AssignConductingPlateBoundary

Creates a conducting plate thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Plate** > **Conducting**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Faces>	list	Faces included in the boundary condition assignment
	<NAME>	string	LowSide
	<Radiate>	string	LowSide: "True" to enable low side for radiation or

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

			"False" to disable it
	<RadiateTo>	string	LowSide: AllObjects or RefTemperature
	<Ref. Temperature>	string	LowSide: Reference temperature and unit
	<View Factor>	string	LowSide: View factor
	<Surface Material>	string	LowSide: Surface material
	<NAME>	string	HighSide
	<Radiate>	string	HighSide: "True" to enable high side for radiation or "False" to disable it
	<RadiateTo - High>	string	HighSide: AllObjects or RefTemperature - High
	<Ref. Temperature - High>	string	HighSide: Reference temperature and unit
	<View Factor - High>	string	HighSide: View factor
	<Surface Material - High>	string	HighSide: Surface material
	<Thermal Specification>	string	Thickness, Conductance, Thermal Resistance, or Thermal Impedance
	<Thickness>	string	Thickness and unit
	<Solid Material>	string	Solid material
	<Conductance>	string	Conductance and unit
	<Thermal Resistance>	string	Thermal resistance and unit
	<Thermal Impedance>	string	Thermal impedance and unit
	<Total Power>	string	Total power and unit
	<Shell Conduction>	bool	"True" to enable shell conduction or "False" to disable it
Return Value	None		

Python Syntax	AssignConductingPlateBoundary (*<NAME>*, *<Faces>*, *<NAME>*, *<Radiate>*, *<RadiateTo>*, *<Ref. Temperature>*, *<View Factor>*, *<Surface Material>*, *<NAME>*, *<Radiate>*, *<RadiateTo - High>*, *<Ref. Temperature - High>*, *<View Factor - High>*, *<Surface Material - High>*, *<Thermal Specification>*, *<Thickness>*, *<Solid Material>*, *<Conductance>*, *<Thermal Resistance>*, *<Thermal Impedance>*, *<Total Power>*, *<Shell Conduction>*)
Python Example	`oModule.AssignConductingPlateBoundary(`

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

```
        [
                "NAME:ConductingPlate1",
                "Faces:="               , [7],
                [
                        "NAME:LowSide",
                        "Radiate:="             , True,
                        "RadiateTo:="           , "RefTemperature",
                        "Ref. Temperature:="    , "0",
                        "View Factor:="         , "0",
                        "Surface Material:="    , "Steel-oxidised-surfac
                ],
                [
                        "NAME:HighSide",
                        "Radiate:="             , True,
                        "RadiateTo - High:="    , "RefTemperature - High
                        "Ref. Temperature - High:=", "0",
                        "View Factor - High:="  , "0",
                        "Surface Material - High:=", "Steel-oxidised-sur
                ],
                "Thermal Specification:=", "Thickness",
                "Thickness:="           , "1mm",
                "Solid Material:="      , "Al-Extruded",
                "Conductance:="         , "0W_per_Cel",
                "Thermal Resistance:="  , "0Kel_per_W",
                "Thermal Impedance:="   , "0celm2_per_w",
                "Total Power:="         , "0W",
                "Shell Conduction:="    , False
        ])
```

AssignResistanceBoundary

Creates a resistance thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Resistance**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Objects>	list	Objects included in the boundary condition assignment
	<Pressure Loss Model>	string	Device/Approach, Power Law, or Loss Curve
	<Laminar Flow>	bool	"True" to enabled laminar flow or 'False' to disable it
	<Thermal Power>	string	Resistance power value and unit
	<Thermal Power>	string	Resistance power value and unit
	<Fluid Material>	string	Fluid material

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

	<Linear X Coefficient>	string	Linear X direction loss coefficient and unit
	<Linear Y Coefficient>	string	Linear Y direction loss coefficient and unit
	<Linear Z Coefficient>	string	Linear Z direction loss coefficient and unit
	<Quadratic X Coefficient>	string	Quadratic X direction loss coefficient
	<Quadratic Y Coefficient>	string	Quadratic Y direction loss coefficient
	<Quadratic Z Coefficient>	string	Quadratic Z direction loss coefficient
	<Linear X Free Area Ratio>	string	Linear X direction free area ratio
	<Linear Y Free Area Ratio>	string	Linear Y direction free area ratio
	<Linear Z Free Area Ratio>	string	Linear Z direction free area ratio
	<Quadratic X Free Area Ratio>	string	Quadratic X direction free area ratio
	<Quadratic Y Free Area Ratio>	string	Quadratic Y direction free area ratio
	<Quadratic Z Free Area Ratio>	string	Quadratic Z direction free area ratio
	<Power Law Coefficient>	string	Power law coefficient
	<Power Law Exponent>	string	Power law exponent
	<NAME>	string	Pressure Loss Curve X
	<NAME>	string	Dataset units
	<X>	list	List of X data set values
	<Y>	list	List of Y data set values
	<NAME>	string	Pressure Loss Curve Y
	<NAME>	string	Dataset units
	<X>	list	List of X data set values
	<Y>	list	List of Y data set values
	<NAME>	string	Pressure Loss Curve Z
	<NAME>	string	Dataset units
	<X>	list	List of X data set values
	<Y>	list	List of Y data set values
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	AssignResistanceBoundary (<*NAME*>, <*Objects*>, <*Pressure Loss Model*>, <*Laminar Flow*>, <*Thermal Power*>, <*Thermal Power*>, <*Fluid Material*>, <*Linear X Coefficient*>, <*Linear Y Coefficient*>, <*Linear Z Coefficient*>, <*Quadratic X Coefficient*>, <*Quadratic Y Coefficient*>, <*Quadratic Z Coefficient*>, <*Linear X Free Area Ratio*>, <*Linear Y Free Area Ratio*>, <*Linear Z Free Area Ratio*>, <*Quadratic X Free Area Ratio*>, <*Quadratic Y Free Area Ratio*>, <*Quadratic Z Free Area Ratio*>, <*Power Law Coefficient*>, <*Power Law Exponent*>, <*NAME*>, <*NAME*>, <*X*>, <*Y*>, <*NAME*>, <*NAME*>, <*X*>, <*Y*>, <*NAME*>, <*NAME*>, <*X*>, <*Y*>)

Python Example	`oModule.AssignResistanceBoundary(`
	`[`
	`"NAME:Resistance1",`
	`"Objects:=" , ["Region"],`
	`"Pressure Loss Model:=" , "Device/Approach",`
	`"Laminar Flow:=" , False,`
	`"Thermal Power:=" , "5W",`
	`"Thermal Power:=" , "5W",`
	`"Fluid Material:=" , "air",`
	`"Linear X Coefficient:=", "0m_per_sec",`
	`"Linear Y Coefficient:=", "0m_per_sec",`
	`"Linear Z Coefficient:=", "0m_per_sec",`
	`"Quadratic X Coefficient:=", "0",`
	`"Quadratic Y Coefficient:=", "0",`
	`"Quadratic Z Coefficient:=", "0",`
	`"Linear X Free Area Ratio:=", "0.8",`
	`"Linear Y Free Area Ratio:=", "0.8",`
	`"Linear Z Free Area Ratio:=", "0.8",`
	`"Quadratic X Free Area Ratio:=", "1",`
	`"Quadratic Y Free Area Ratio:=", "1",`
	`"Quadratic Z Free Area Ratio:=", "1",`
	`"Power Law Coefficient:=", "1",`
	`"Power Law Exponent:=" , "1",`
	`[`
	`"NAME:Pressure Loss Curve X",`
	`[`
	`"NAME:DimUnits",`
	`"",`
	`""`
	`],`
	`"X:=" , ["0","1","2"],`
	`"Y:=" , ["0","1","2"]`
	`],`
	`[`
	`"NAME:Pressure Loss Curve Y",`
	`[`
	`"NAME:DimUnits",`
	`"",`

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

```
                                        ""
                                ],
                                "X:="                   , ["0","1","2"],
                                "Y:="                   , ["0","1","2"]
                        ],
                        [
                                "NAME:Pressure Loss Curve Z",
                                [
                                        "NAME:DimUnits",
                                        "",
                                        ""
                                ],
                                "X:="                   , ["0","1","2"],
                                "Y:="                   , ["0","1","2"]
                        ]
                ])
```

AssignSourceBoundary

Creates a source thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Source**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Faces>	list	Faces included in the boundary condition assignment
	<Thermal Condition>	string	Total Power, Surface flux, or Fixed Temperature
	<Total Power>	string	Source power value and unit
	<Surface Heat>	string	Source surface heat value and unit
	<Temperature>	string	Source temperature value and unit
	<NAME>	string	Radiation
	<Radiate>	string	"True" to enable radiation or "False" to disable it
	<Voltage/Current - Enabled>	bool	"True" to enable voltage or current or "False" to disable it
	<Voltage/Current Option>	string	Current or Voltage
	<Current>	string	Current value and unit
	<Voltage>	string	Voltage vale and unit
	<RadiateTo>	string	AllObjects or RefTemperature
	<Ref. Temperature>	string	Reference temperature and unit
	<View Factor>	string	View factor

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

	<*Surface Material*>	string	Surface material
Return Value	None		

Python Syntax	AssignSourceBoundary (<*NAME*>, <*Faces*>, <*Thermal Condition*>, <*Total Power*>, <*Surface Heat*>, <*Temperature*>, <*NAME*>, <*Radiate*>, <*RadiateTo*>, <*Ref. Temperature*>, <*View Factor*>, <*Surface Material*>)

Python Example	`oModule.AssignSourceBoundary(`
	`  [`
	`  "NAME:Source1",`
	`  "Faces:=" , [7],`
	`  "Thermal Condition:=" , "Total Power",`
	`  "Total Power:=" , "5W",`
	`  "Surface Heat:=" , "0irrad_W_per_m2",`
	`  "Temperature:=" , "AmbientTemp",`
	`  [`
	`  "NAME:Radiation",`
	`  "Radiate:=" , True,`
	`  "RadiateTo:=" , "RefTemperature",`
	`  "Ref. Temperature:=" , "0",`
	`  "View Factor:=" , "1",`
	`  "Surface Material:=" , "Steel-oxidised-surfa`
	`  ]`
	`  "Voltage/Current - Enabled:=", True,`
	`  "Voltage/Current Option:=", "Current",`
	`  "Current:=" , "0A",`
	`  "Voltage:=" , "0V"`
	`  ])`

AssignStationaryWallBoundary

Creates a stationary wall thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Wall** > **Stationary**		
Parameters	Name	Type	Description
	<*NAME*>	string	Boundary condition name
	<*Faces*>	list	Faces included in the boundary condition assignment
	<*Thickness*>	string	Wall thickness and unit
	<*Solid Material*>	string	Solid material
	<*External Material*>	string	External material
	<*Internal Material*>	string	Internal material
	<*External Condition*>	string	Heat flux, Temperature, or

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

			Heat Transfer Coefficient
	<Heat Flux>	string	Wall heat flux and unit
	<Temperature>	string	Wall temperature and unit
	<Heat Transfer Coefficient>	string	Wall heat transfer coefficient and unit
	<Reference Temperature>	string	Reference temperature and unit
	<NAME>	string	Heat Transfer Data
	<Heat Transfer Correlation>	bool	"True" to enable heat transfer correlation or "False" to disable it
	<Heat Transfer Convection Type>	string	Forced or Natural
	<NAME>	string	Radiation
	<Radiate>	bool	"True" to enable radiation or "False" to disable it
	<Shell Conduction>	bool	"True" to enable shell conduction or "False" to disable it
	<Slip Flow Boundary>	bool	"True" to enable skip flow boundary or "False" to disable it
Return Value	None		

Python Syntax	AssignStationaryWallBoundary (*<NAME>*, *<Faces>*, *<Thickness>*, *<Solid Material>*, *<External Material>*, *<Internal Material>*, *<External Condition>*, *<Heat Flux>*, *<Temperature>*, *<Heat Transfer Coefficient>*, *<Reference Temperature>*, *<NAME>*, *<Heat Transfer Correlation>*, *<Heat Transfer Convection Type>*, *<NAME>*, *<Radiate>*, *<Shell Conduction>*)

Python Example	`oModule.AssignStationaryWallBoundary(`
	`[`
	`"NAME:StationaryWall1",`
	`"Faces:=" , [12],`
	`"Thickness:=" , "2mm",`
	`"Solid Material:=" , "Al-Extruded",`
	`"External Material:=" , "Steel-oxidised-surface",`
	`"Internal Material:=" , "Steel-oxidised-surface",`
	`"External Condition:=" , "Heat Flux",`
	`"Heat Flux:=" , "5uW_per_m2",`
	`"Temperature:=" , "AmbientTemp",`
	`"Heat Transfer Coefficient:=", "0w_per_m2kel",`
	`"Reference Temperature:=", "AmbientTemp",`
	`[`
	`"NAME:Heat Transfer Data",`

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

	"Heat Transfer Correlation:=", False, "Heat Transfer Convection Type:=", "Forced Conve], ["NAME:Radiation", "Radiate:=" , False], "Shell Conduction:=" , False])

AssignSymmetryWallBoundary

Creates a symmetry wall thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Wall** > **Symmetry**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Faces>	list	Faces included in the boundary condition assignment
Return Value	None		

Python Syntax	AssignSymmetryWallBoundary (*<NAME>*, *<Faces>*)
Python Example	oModule.AssignSymmetryWallBoundary(["NAME:SymmetryWall1", "Faces:=" , [12]])

AssignEMLoss

Creates an EM loss thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **EM Loss**		
Parameters	Name	Type	Description
	<NAME>	string	Boundary condition name
	<Objects>	list	Objects included in the boundary condition assignment
	<Project>	string	Source design's project name

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

	<Product>	string	Source design's design type
	<Design>	int	Source design's name
	<Soln>	bool	Source design's solution name
	<NAME>	string	Params
	<ForceSourceToSolve>	bool	"True" to solve the source project or "False" to not solve it
	<PreservePartnerSoln>	bool	"True" to save the source design results or "False" to discard them
	<PathRelativeTo>	string	TargetProject or SourceProduct
	<Intrinsics>	list	List of frequencies
	<IntrinsicsHarmonicLoss>	list	Q3D only: list of frequency values with units
	<HarmonicLossSweepCoupling>	bool	Q3D only: True or False
	<Q3DEMLossType>	string	Q3D only: HarmonicLoss for harmonic loss coupling, DCVolOrACSurfLoss for DC volume or AC surface coupling, or ContactResistanceLoss for DC contact resistance loss coupling
	<SurfaceOnly>	list	List of surfaces
Return Value	None		

Python Syntax	AssignEMLoss (*<NAME>*, *<Objects>*, *<Project>*, *<Product>*, *<Design>*, *<Soln>*, *<NAME>*, *<ForceSourceToSolve>*, *<PreservePartnerSoln>*, *<PathRelativeTo>*, *<Intrinsics>*, *<SurfaceOnly>*)

Python Example	`oModule.AssignEMLoss(`
	`  [`
	`  "NAME:EMLoss1",`
	`  "Objects:=" , ["Stock_0","Stock_1","Stock_2"`
	`  "Project:=" , "This Project*",`
	`  "Product:=" , "ElectronicsDesktop",`
	`  "Design:=" , "Maxwell3DDesign1",`
	`  "Soln:=" , "Setup1 : LastAdaptive",`
	`  [`
	`  "NAME:Params"`
	`  ],`
	`  "ForceSourceToSolve:=" , False,`
	`  "PreservePartnerSoln:=" , False,`
	`  "PathRelativeTo:=" , "TargetProject",`
	`  "Intrinsics:=" , ["200Hz"],`

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

	"IntrinsicsHarmonicLoss:=", ["0Hz","1Hz","10Hz"], "HarmonicLossSweepCoupling:=", False, "Q3DEMLossType:=" , "ContactResistanceLoss", "SurfaceOnly:=" , []])

AssignBlowerBoundary

Creates a blower thermal boundary condition in an Icepak design.

UI Access	**Icepak** > **Thermal** > **Assign** > **Blower**
Parameters	

Name	Type	Description
<NAME>	string	Boundary condition name
<Faces>	list	Faces included in the boundary condition assignment
<NAME>	string	DimUnits
<X>	list	Flow curve dataset x values
<Y>	list	Flow curve dataset y values
<Blower Type>	int	Type 1 or Type 2
<Fan Blade Angle>	string	Type 1 only: angle of the blower fan blade
<Blade RPM>	string	Type 1 only: rotations per minute value
<Exhaust Exit Angle>	string	Type 2 only: blower exhaust exit angle value and unit
<Blower Power>	string	Blower power value and unit
<InletFace>	list	Inlet face

Return Value	None

Python Syntax	AssignBlowerBoundary (*<NAME>*, *<Faces>*, *<NAME>*, *<X>*, *<Y>*, *<Blower Type>*, *<Fan Blade Angle>*, *<Blade RPM>*, *<Blower Power>*, *<InletFace>*)
Python Example	oModule.AssignBlowerBoundary(["NAME:Blower1", "Faces:=" , [95,94,93], ["NAME:DimUnits", "m_per_sec", "n_per_meter_sq"],

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

```
                "X:="                   , ["0","1","2"],
                "Y:="                   , ["0","1","2"],
                "Blower Type:="         , "Type 1",
                "Fan Blade Angle:="     , "0rad",
                "Blade RPM:="           , "0",
                "Blower Power:="        , "0W",
                "InletFace:="           , [95,94]
        ])
```

AssignHeatExchangerBoundary

Creates a heat exchanger thermal boundary condition in an Icepak design.

UI Access	**cepak|** > **Thermal** > **Assign** > **Heat Exchanger**

Parameters

Name	Type	Description
<NAME>	string	Boundary condition name
<Faces>	list	Faces included in the boundary condition assignment
<Pressue Loss type>	string	Constant or Polynomial
<Constant Pressure Loss>	string	Constant pressure loss value
<Polynomial Pressure Loss>	list	Polynomial pressure loss values in brackets separated by commas. For example, [1,2].
<Thermal Condition>	string	Heat Flux or Heat Transfer Coefficient
<Heat flux>	string	Heat flux value and unit
<Heat Transfer Coefficient Type>	string	Constant or Polynomial
<Constant HTC>	string	Constant heat transfer coefficient value and unit
<Polynomial HTC>	list	Polynomial heat transfer coefficent values in brackets separated by commas. For example, [1,2].
<External Temperature>	string	AmbientTemp or temperature value and unit
<Allow Merge>	bool	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	AssignHeatExchangerBoundary (<*NAME*>, <*Faces*>, <*Pressue Loss type*>, <*Constant Pressure Loss*>, <*Polynomial Pressure Loss*>, <*Thermal Condition*>, <*Heat flux*>, <*Heat Transfer Coefficient Type*>, <*Constant HTC*>, <*Polynomial HTC*>, <*External Temperature*>, <*Allow Merge*>)

Python Example	`oModule.AssignHeatExchangerBoundary(`
	`[`
	`"NAME:HeatExchanger1",`
	`"Faces:=" , [38],`
	`"Pressue Loss type:=" , "Constant",`
	`"Constant Pressure Loss:=", "1",`
	`"Polynomial Pressure Loss:=", [1,1],`
	`"Thermal Condition:=" , "Heat Flux",`
	`"Heat flux:=" , "2mW_per_m2",`
	`"Heat Transfer Coefficient Type:=", "Constant",`
	`"Constant HTC:=" , "0w_per_m2kel",`
	`"Polynomial HTC:=" , [1,1],`
	`"External Temperature:=", "AmbientTemp",`
	`"Allow Merge:=" , False`
	`])`

AssignSolarLoading

Assigns solar loading to an object.

UI Access	**Icepak** > **Solar Loading** > **Assign Solar Loading**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME*></td><td>string</td><td>Solar loading name</td></tr><tr><td><*Objects*></td><td>list</td><td>Objects included in the solar loading assignment</td></tr><tr><td><*Solar Loading*></td><td>bool</td><td>True or False</td></tr></table>
Return Value	None

Python Syntax	AssignSolarLoading (<*NAME*>, <*Objects*>, <*Solar Loading*>)

Python Example	`oModule.AssignSolarLoading(`
	`[`
	`"NAME:SolarLoading1",`
	`"Objects:=" , ["SERIAL_PORT"],`
	`"Solar Loading:=" , True`

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

	])

AssignInitialTemperature

Assigns an initial temperature patch to an object.

UI Access	**Icepak** > **Initial Temperature** > **Assign**		
Parameters	**Name**	**Type**	**Description**
	<NAME>	string	Temperature patch name
	<Objects>	list	Objects included in the temperature patch assignment
	<Initial Temperature>	string	"AmbientTemp" or initial temperature value and unit
Return Value	None		

Python Syntax	AssignInitialTemperature (*<NAME>*, *<Objects>*, *<Initial Temperature>*)

Python Example	`oModule.AssignInitialTemperature(`
	`  [`
	`  "NAME:InitTemp1",`
	`  "Objects:=" , ["Box1"],`
	`  "Initial Temperature:=" , "AmbientTemp"`
	`  ])`

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

12 - Native Component Module Script Commands

Boundary condition commands should be executed by the "InsertNativeComponent" module.

```
oModule = oDesign.GetModule("InsertNativeComponent")
```

The topics for this section include:

[InsertNativeComponent (Fan)]

[InsertNativeComponent (Heatsink)]

[InsertNativeComponent (PCB)]

[InsertNativeComponent (CTM)]

[InsertNativeComponent (Package)]

InsertNativeComponent (Fan)

Creates a fan component in an Icepak design.

UI Access	**Project Manager** > **3D Component** > **Create** > **Fan**		
Parameters	Name	Type	Description
	<NAME>	string	InsertNativeComponentData
	<TargetCS>	string	Coordinate system for fan component
	<SubmodelDefinitionName>	string	Submodel name
	<NAME>	string	ComponentPriorityLists
	<NextUniqueID>	int	ID value
	<MoveBackwards>	bool	True or False
	<DatasetType>	string	ComponentDatasetType
	<NAME>	string	DatasetDefinitions
	<NAME>	string	BasicComponentInfo
	<ComponentName>	string	Name of fan component
	<Company>	string	Name of manufacturer
	<Company URL>	string	Manufacturer website URL
	<Model Number>	string	Manufacturer model number
	<Help URL>	string	Manufacturer online help URL
	<Version>	list	Fan component version number
	<Notes>	string	Notes on fan component
	<IconType>	string	Fan
	<NAME>	string	GeometryDefinitionParameters
	<NAME>	string	VariableOrders

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	DesignDefinitionParameters
	<NAME>	string	VariableOrders
	<NAME>	string	MaterialDefinitionParameters
	<NAME>	string	VariableOrders
	<MapInstanceParameters>	string	DesignVariable
	<UniqueDefinitionIdentifier>	int	ID number
	<OriginFilePath>	string	File path
	<IsLocal>	bool	True or False
	<ChecksumString>	string	Checksum string text
	<ChecksumHistory>	list	Checksum history text
	<VersionHistory>	list	Version history text
	<NAME>	string	NativeComponentDefinitionProvider
	<Type>	string	Fan
	<Unit>	string	Length unit
	<ModelAs>	string	2D or 3D
	<Shape>	string	Circular, Rectangular, or Polygon
	<MovePlane>	string	XY, YZ, or ZX
	<Radius>	string	Fan radius and unit
	<HubRadius>	string	Fan hub radius and unit
	<CaseSide>	bool	3D Fans only: Fan case side length and unit
	<FlowDirChoice>	string	Normalpositive or NormalNegative
	<FlowType>	string	Curve, FixedVolumetric, or FixedMassFlow
	<SwirlType>	string	Magnitude or RPM
	<NAME>	string	DimUnits
	<X>	string	X coordinate values
	<Y>	list	Y coordinate values
	<IntakeTemp>	string	Fan intake temperature
	<Swirl>	string	Fan swirl value
	<OperatingRPM>	string	Fan operating RPM
	<NAME>	string	InstanceParameters
	<GeometryParameters>	string	Fan geometry parameter values
	<MaterialParameters>	string	Fan material parameter values
	<DesignParameters>	string	Fan design parameter values
Return Value	None		

Python Syntax	InsertNativeComponent (*<NAME>*, *<TargetCS>*, *<SubmodelDefinitionName>*, *<NAME>*, *<NextUniqueID>*, *<MoveBackwards>*, *<DatasetType>*, *<NAME>*,

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

	<*NAME*>, <*ComponentName*>, <*Company*>, <*Company URL*>, <*Model Number*>, <*Help URL*>, <*Version*>, <*Notes*>, <*IconType*>, <*NAME*>, <*NAME*>, <*NAME*>, <*NAME*>, <*NAME*>, <*NAME*>, <*MapInstanceParameters*>, <*UniqueDefinitionIdentifier*>, <*OriginFilePath*>, <*IsLocal*>, <*ChecksumString*>, <*ChecksumHistory*>, <*VersionHistory*>, <*NAME*>, <*Type*>, <*Unit*>, <*ModelAs*>, <*Shape*>, <*MovePlane*>, <*Radius*>, <*HubRadius*>, <*CaseSide*>, <*FlowDirChoice*>, <*FlowType*>, <*SwirlType*>, <*NAME*>, <*X*>, <*Y*>, <*IntakeTemp*>, <*Swirl*>, <*OperatingRPM*>, <*NAME*>, <*GeometryParameters*>, <*MaterialParameters*>, <*DesignParameters*>)

Python Example	`oEditor.InsertNativeComponent(`
	`  [`
	`  "NAME:InsertNativeComponentData",`
	`  "TargetCS:=" , "Global",`
	`  "SubmodelDefinitionName:=", "Fan1",`
	`  [`
	`  "NAME:ComponentPriorityLists"`
	`  ],`
	`  "NextUniqueID:=" , 0,`
	`  "MoveBackwards:=" , False,`
	`  "DatasetType:=" , "ComponentDatasetType",`
	`  [`
	`  "NAME:DatasetDefinitions"`
	`  ],`
	`  [`
	`  "NAME:BasicComponentInfo",`
	`  "ComponentName:=" , "Fan1",`
	`  "Company:=" , "",`
	`  "Company URL:=" , "",`
	`  "Model Number:=" , "",`
	`  "Help URL:=" , "",`
	`  "Version:=" , "1.0",`
	`  "Notes:=" , "",`
	`  "IconType:=" , "Fan"`
	`  ],`
	`  [`
	`  "NAME:GeometryDefinitionParameters",`
	`  [`
	`  "NAME:VariableOrders"`
	`  ]`
	`  ],`
	`  [`
	`  "NAME:DesignDefinitionParameters",`
	`  [`
	`  "NAME:VariableOrders"`
	`]

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

```
                ],
                [
                        "NAME:MaterialDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                "MapInstanceParameters:=", "DesignVariable",
                "UniqueDefinitionIdentifier:=", "bb86a759-8aeb-48ba-b1a6-
b07ac395e931",
                "OriginFilePath:="      , "",
                "IsLocal:="             , False,
                "ChecksumString:="      , "",
                "ChecksumHistory:="     , [],
                "VersionHistory:="      , [],
                [
                        "NAME:NativeComponentDefinitionProvider",
                        "Type:="                , "Fan",
                        "Unit:="                , "mm",
                        "ModelAs:="             , "2D",
                        "Shape:="               , "Circular",
                        "MovePlane:="           , "XY",
                        "Radius:="              , "0.1mm",
                        "HubRadius:="           , "0mm",
                        "CaseSide:="            , True,
                        "FlowDirChoice:="       , "NormalPositive",
                        "FlowType:="            , "Curve",
                        "SwirlType:="           , "Magnitude",
                        [
                                "NAME:DimUnits",
                                "",
                                ""
                        ],
                        "X:="                   , ["0","0.01"],
                        "Y:="                   , ["30","0"],
                        "IntakeTemp:="          , "AmbientTemp",
                        "Swirl:="               , "0",
                        "OperatingRPM:="        , "0"
                ],
                [
                        "NAME:InstanceParameters",
                        "GeometryParameters:="  , "",
                        "MaterialParameters:="  , "",
                        "DesignParameters:="    , ""
                ]
        ])
```


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

InsertNativeComponent (Heatsink)

Creates a heatsink component in an Icepak design.

UI Access	**Project Manager** > **3D Component** > **Create** > **Heatsink**		
Parameters	Name	Type	Description
	<NAME>	string	InsertNativeComponentData
	<TargetCS>	string	Coordinate system for heatsink component
	<SubmodelDefinitionName>	string	Submodel name
	<NAME>	string	ComponentPriorityLists
	<NextUniqueID>	int	ID value
	<MoveBackwards>	bool	True or False
	<DatasetType>	string	ComponentDatasetType
	<NAME>	string	DatasetDefinitions
	<NAME>	string	BasicComponentInfo
	<ComponentName>	string	Name of heatsink component
	<Company>	string	Name of manufacturer
	<Company URL>	string	Manufacturer website URL
	<Model Number>	string	Manufacturer model number
	<Help URL>	string	Manufacturer online help URL
	<Version>	list	Heatsink component version number
	<Notes>	string	Notes on heatsink component
	<IconType>	string	Heatsink
	<NAME>	string	GeometryDefinitionParameters
	<NAME>	string	VariableOrders
	<NAME>	string	DesignDefinitionParameters
	<NAME>	string	VariableOrders
	<NAME>	string	MaterialDefinitionParameters
	<NAME>	string	VariableOrders
	<MapInstanceParameters>	string	DesignVariable
	<UniqueDefinitionIdentifier>	string	ID number
	<OriginFilePath>	string	File path
	<IsLocal>	bool	True or False
	<ChecksumString>	string	Checksum string text
	<ChecksumHistory>	list	Checksum history text
	<VersionHistory>	list	Version history text
	<NAME>	string	NativeComponentDefinitionProvider
	<Type>	string	Heatsink
	<Unit>	string	Length unit
	<MovePlane>	string	XY, YZ, or ZX

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

	<FinType>	string	Extruded, CrossCutExtrusion, or CylindricalPin
	<FlowDirection>	string	Length or Width
	<BaseLength>	string	Length of heatsink base and unit
	<BaseWidth>	string	Width of heatsink base and unit
	<BaseHeight>	string	Height of heatsink base and unit
	<OverallHeight>	string	Overall height of heatsink and unit
	<FinCount>	string	Number of heatsink fins
	<FinThickness>	string	Fin thickness and unit
	<FinOffsetOnLength>	string	Fin offset on length and unit
	<FinOffsetOnWidth>	string	Fin offset on width and unit
	<BaseMaterial>	string	Material of heatsink base
	<FinMaterial>	string	Material of heatsink fins
	<BaseSurfaceMaterial>	string	Surface material of heatsink base
	<FinSurfaceMaterial>	string	Surface material of heatsink fins
	<NAME>	string	InstanceParameters
	<GeometryParameters>	string	Heatsink geometry parameter values
	<MaterialParameters>	string	Heatsink material parameter values
	<DesignParameters>	string	Heatsink design parameter values
Return Value	placeholder		

Python Syntax	InsertNativeComponent (*<NAME>*, *<TargetCS>*, *<SubmodelDefinitionName>*, *<NAME>*, *<NextUniqueID>*, *<MoveBackwards>*, *<DatasetType>*, *<NAME>*, *<NAME>*, *<ComponentName>*, *<Company>*, *<Company URL>*, *<Model Number>*, *<Help URL>*, *<Version>*, *<Notes>*, *<IconType>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<MapInstanceParameters>*, *<UniqueDefinitionIdentifier>*, *<OriginFilePath>*, *<IsLocal>*, *<ChecksumString>*, *<ChecksumHistory>*, *<VersionHistory>*, *<NAME>*, *<Type>*, *<Unit>*, *<MovePlane>*, *<FinType>*, *<FlowDirection>*, *<BaseLength>*, *<BaseWidth>*, *<BaseHeight>*, *<OverallHeight>*, *<FinCount>*, *<FinThickness>*, *<FinOffsetOnLength>*, *<FinOffsetOnWidth>*, *<BaseMaterial>*, *<FinMaterial>*, *<BaseSurfaceMaterial>*, *<FinSurfaceMaterial>*, *<NAME>*, *<GeometryParameters>*, *<MaterialParameters>*, *<DesignParameters>*)

Python Example	`oEditor.InsertNativeComponent(`
	`  [`
	`  "NAME:InsertNativeComponentData",`
	`  "TargetCS:=" , "Global",`
	`  "SubmodelDefinitionName:=", "Heatsink1",`
	`  [`

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

```
                        "NAME:ComponentPriorityLists"
                ],
                "NextUniqueID:="        , 0,
                "MoveBackwards:="       , False,
                "DatasetType:="         , "ComponentDatasetType",
                [
                        "NAME:DatasetDefinitions"
                ],
                [
                        "NAME:BasicComponentInfo",
                        "ComponentName:="       , "Heatsink1",
                        "Company:="             , "",
                        "Company URL:="         , "",
                        "Model Number:="        , "",
                        "Help URL:="            , "",
                        "Version:="             , "1.0",
                        "Notes:="               , "",
                        "IconType:="            , "HeatSink"
                ],
                [
                        "NAME:GeometryDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:DesignDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:MaterialDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                "MapInstanceParameters:=", "DesignVariable",
                "UniqueDefinitionIdentifier:=", "90564449-0b94-41d4-af0c-
cdde6b4c1c26",
                "OriginFilePath:="      , "",
                "IsLocal:="             , False,
                "ChecksumString:="      , "",
                "ChecksumHistory:="     , [],
                "VersionHistory:="      , [],
                [
                        "NAME:NativeComponentDefinitionProvider",
```


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

```
seLength:="                , "0.2mm",
                        "BaseWidth:="           , "0.2mm",
                        "BaseHeight:="          , "0.025mm",
                        "OverallHeight:="       , "0.1mm",
                        "FinCount:="            , "4",
                        "FinThickness:="        , "0.02mm",
                        "FinOffsetOnLength:="   , "0mm",
                        "FinOffsetOnWidth:="    , "0mm",
                        "BaseMaterial:="        , "Al-Extruded',
                        "FinMaterial:="         , "Al-Extruded',
                        "BaseSurfaceMaterial:=" , "Steel-oxidised-surface"
                        "FinSurfaceMaterial:="  , "Steel-oxidised-surface"
                ],
                [
                        "NAME:InstanceParameters",
                        "GeometryParameters:="  , "",
                        "MaterialParameters:="  , "",
                        "DesignParameters:="    , ""
                ]
        ])
```

InsertNativeComponent (PCB)

Creates a PCB native component in an Icepak design.

UI Access	**Project Manager** > **3D Component** > **Create** > **PCB**

Parameters

Name	Type	Description
<NAME>	string	Component name
<TargetCS>	string	Component coordinate system
<SubmodelDefinitionName>	string	Component submodel name
<NAME>	string	ComponentPrioritylists
<NextUniqueID>	int	Identification number
<MoveBackwards>	bool	True or False
<DatasetType>	string	ComponentDatasetType
<NAME>	string	DatasetDefinitions
<NAME>	string	BasicComponentInfo
<ComponentName>	string	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.

	g	
<Company>	string	Component company name
<Company URL>	string	Component company website address
<Model Number>	string	Component model number
<Help URL>	string	Component online help website address
<Version>	list	Component version
<Notes>	string	Component notes
<IconType>	string	PCB
<NAME>	string	GeometryDefinitionParameters
<NAME>	string	VariableOrders
<NAME>	string	DesignDefinitionParameters
<NAME>	string	VariableOrders
<NAME>	string	MaterialDefinitionParameters
<NAME>	string	VariableOrders
<MapInstanceParameters>	string	DesignVariable
<UniqueDefinitionIdentifier>	int	Unique ID number
<OriginFilePath>	string	Component file path
<IsLocal>	bool	True or False
<ChecksumString>	string	Checksum
<ChecksumHistory>	list	Checksum history
<VersionHistory>	list	Version history
<NAME>	string	NativeComponentDefinitionProvider
<Type>	string	PCB
<Unit>	string	Unit type
<MovePlane>	string	XY, YZ, or XZ
<Use3DLayoutExtents>	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.

<ModelLayersLumped>	bool	True or False
<ExtentsType>	string	Polygon or BoundingBox
<OutlinePolygon>	int	Outline polygon name
<BoardCutoutMaterial>	string	Material for board cutouts
<PartsChoice>	int	0, 1, or 2
<CreateTopSolderballs>	bool	True or False
<TopConnectorType>	string	Solderbump or Bondwire
<BondwireMaterial>	string	Material for bondwires
<BondwireDiameter>	string	Bondwire diameter value and unit
<TopSolderballSModelType>	string	Solderbump only: SbBlock, SbCylinder, or SbLumped
<CreateBottomSolderballs>	bool	True or False
<BottomSolderballSModelType>	string	Solderbump only: SbBlock, SbCylinder, or SbLumped
<Filters>	list	["Cap","Height","HeightExclude2D","Ind","Power","Res"]
<CreateFilteredAsNonModel>	bool	True or False
<FootPrint>	string	Footprint size and unit
<NAME>	string	definitionOverridesMap
<NAME>	string	instanceOverridesMap
<NAME>	string	oneOverrideBlk
<overrideName>	string	Overriden instance name
<NAME>	string	overrideProps
<isFiltered>	bool	True or False
<isOverridePower>	bool	True or False
<isOverrideThetaJb>	bool	True or False
<isOverrideThetaJc>	bool	True or False
<powerOverride>	string	Power override value
<HighSurfThickness>	string	High side surface thickness value and unit

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

	g	
<LowSurfThickness>	string	Low side surface thickness value and unit
<InternalLayerThickness>	string	Internal layer thickness value 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
<ReplaceDevices>	bool	True or False
<IncludeMCAD>	bool	True or False
<Resolution>	bool	True or False
<NAME>	string	LowSide
<Radiate>	bool	True or False
<RadiateTo>	string	AllObjects or ReferenceTemperature
<Surface Material>	string	Surface material name
<NAME>	string	HighSide
<Radiate>	bool	True or False
<UseThermalLink>	bool	True or False
<CustomResolution>	bool	True or False
<Power>	string	Power value and unit
<NAME>	string	DefnLink
<Project>	string	Project name or This Project*
<Product>	string	ElectronicsDesktop
<Design>	string	Design name
<Soln>	bool	True or False
<NAME>	string	Params
<ForceSourceToSolve>	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.

	<PreservePartnerSoln>	bool	True or False
	<PathRelativeTo>	string	TargetProject
	<NAME>	string	InstanceParameters
	<GeometryParameters>	string	Geometry parameters list
	<MaterialParameters>	string	Material parameters list
	<DesignParameters>	string	Design parameters list
Return Value	None		

Python Syntax	InsertNativeComponent (*<NAME>*, *<TargetCS>*, *<SubmodelDefinitionName>*, *<NAME>*, *<NextUniqueID>*, *<MoveBackwards>*, *<DatasetType>*, *<NAME>*, *<NAME>*, *<ComponentName>*, *<Company>*, *<Company URL>*, *<Model Number>*, *<Help URL>*, *<Version>*, *<Notes>*, *<IconType>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<MapInstanceParameters>*, *<UniqueDefinitionIdentifier>*, *<OriginFilePath>*, *<IsLocal>*, *<ChecksumString>*, *<ChecksumHistory>*, *<VersionHistory>*, *<NAME>*, *<Type>*, *<Unit>*, *<MovePlane>*, *<Use3DLayoutExtents>*, *<ExtentsType>*, *<OutlinePolygon>*, *<PartsChoice>*, *<Board>*, *<Library>*, *<Control>*, *<Filters>*, *<CreateFilteredAsNonModel>*, *<NAME>*, *<NAME>*, *<NAME>*, *<overrideName>*, *<NAME>*, *<isFiltered>*, *<isOverridePower>*, *<isOverrideThetaJb>*, *<isOverrideThetaJc>*, *<powerOverride>*, *<NAME>*, *<overrideName>*, *<NAME>*, *<isFiltered>*, *<isOverridePower>*, *<isOverrideThetaJb>*, *<isOverrideThetaJc>*, *<HighSurfThickness>*, *<LowSurfThickness>*, *<InternalLayerThickness>*, *<NumInternalLayer>*, *<HighSurfaceCopper>*, *<LowSurfaceCopper>*, *<InternalLayerCopper>*, *<TraceMaterial>*, *<SubstrateMaterial>*, *<CreateBoard>*, *<ModelBoardAsRect>*, *<ModelDeviceAsRect>*, *<Cutoff>*, *<ReplaceDevices>*, *<IncludeMCAD>*, *<Resolution>*, *<NAME>*, *<Radiate>*, *<RadiateTo>*, *<Surface Material>*, *<NAME>*, *<Radiate>*, *<UseThermalLink>*, *<CustomResolution>*, *<Power>*, *<NAME>*, *<Project>*, *<Product>*, *<Design>*, *<Soln>*, *<NAME>*, *<ForceSourceToSolve>*, *<PreservePartnerSoln>*, *<PathRelativeTo>*, *<NAME>*, *<GeometryParameters>*, *<MaterialParameters>*, *<DesignParameters>*)
Python Example	`oEditor.InsertNativeComponent(`

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

```
	[
		"NAME:InsertNativeComponentData",
		"TargetCS:="		, "Global",
		"SubmodelDefinitionName:=", "PCB1",
		[
			"NAME:ComponentPriorityLists"
		],
		"NextUniqueID:="	, 0,
		"MoveBackwards:="	, False,
		"DatasetType:="		, "ComponentDatasetType",
		[
			"NAME:DatasetDefinitions"
		],
		[
			"NAME:BasicComponentInfo",
			"ComponentName:="	, "PCB1",
			"Company:="		, "",
			"Company URL:="		, "",
			"Model Number:="	, "",
			"Help URL:="		, "",
			"Version:="		, "1.0",
			"Notes:="		, "",
			"IconType:="		, "PCB"
		],
		[
			"NAME:GeometryDefinitionParameters",
			[
				"NAME:VariableOrders"
			]
		],
		[
			"NAME:DesignDefinitionParameters",
			[
				"NAME:VariableOrders"
			]
		],
		[
			"NAME:MaterialDefinitionParameters",
			[
				"NAME:VariableOrders"
			]
		],
		"MapInstanceParameters:=", "DesignVariable",
```


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

```
:=", "e0363806-39b6-4de6-8f07-13ac9e2a1bca",
                "OriginFilePath:="      , "",
                "IsLocal:="             , False,
                "ChecksumString:="      , "",
                "ChecksumHistory:="     , [],
                "VersionHistory:="      , [],
                [
                        "NAME:NativeComponentDefinitionProvider",
                        "Type:="                , "PCB",
                        "Unit:="                , "mm",
                        "MovePlane:="           , "XY",
                        "Use3DLayoutExtents:="  , False,
                        "ExtentsType:="         , "Polygon",
                        "OutlinePolygon:="      , "poly_0",
                        "PartsChoice:="         , 1,
                        "Board:="               , "",
                        "Library:="             , "",
                        "Control:="             , "",
                        "Filters:="             , ["HeightExclude2D"],
                        "CreateFilteredAsNonModel:=", False,
                        "FootPrint:="           , "0.1mm2",
                        [
                                "NAME:definitionOverridesMap"
                        ],
                        [
                                "NAME:instanceOverridesMap",
                                [
                                        "NAME:oneOverrideBlk",
                                        "overrideName:="        , "D9",
                                        [
                                                "NAME:overrideProps",
                                                "isFiltered:="          ,
                                                "isOverridePower:="     ,
                                                "isOverrideThetaJb:="   ,
                                                "isOverrideThetaJc:="   ,
                                                "powerOverride:="       ,
                                        ]
                                ],
                                [
                                        "NAME:oneOverrideBlk",
                                        "overrideName:="        , "P1",
                                        [
                                                "NAME:overrideProps",
                                                "isFiltered:="          ,
                                                "isOverridePower:="     ,
                                                "isOverrideThetaJb:="   ,
```


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

```
        "Project:="              , "This Project*",
                                 "Product:="            , "ElectronicsDesk
                                 "Design:="             , "PCB-00278 A 172
                                 "Soln:="               , "",
                                 [
                                         "NAME:Params"
                                 ],
                                 "ForceSourceToSolve:="  , False,
                                 "PreservePartnerSoln:=" , False,
                                 "PathRelativeTo:="      , "TargetProject"
                         ]
                 ],
                 [
                         "NAME:InstanceParameters",
                         "GeometryParameters:="  , "",
                         "MaterialParameters:="  , "",
                         "DesignParameters:="    , ""
                 ]
        ])
```

InsertNativeComponent (CTM)

Creates a CTM native component in an Icepak design.

> **Note:**
>
> When recording a script while editing CTM native component, updates to the source power values are not recorded unless the revised CTM file is imported.

UI Access	**Project Manager** > **3D Component** > **Create** > **CTM**		
Parameters	Name	Type	Description
	<NAME>	string	InsertNativeComponentData
	<NAME>	string	FaceCSParameters
	<KernelVersion>	int	Kernel version
	<PartID>	int	Part identification number
	<NAME>	string	Origin
	<IsAttachedToEntity>	bool	True
	<EntityID>	int	Entity identification number
	<FacetedBodyTriangleIndex>	int	Index value
	<TriangleVertexIndex>	int	Index value
	<PositionType>	string	FaceCenter
	<UParam>	int	UParam value

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

<VParam>	int	UParam value
<XPosition>	string	X position value
<YPosition>	string	Y position value
<ZPosition>	string	Z position value
<MoveToEnd>	bool	False
<FaceID>	int	Face identification number
<NAME>	string	AxisPosn
<IsAttachedToEntity>	bool	True
<EntityID>	int	Entity identification number
<FacetedBodyTri-angleIndex>	int	Index value
<TriangleVertexIndex>	int	Index value
<PositionType>	string	EdgeCenter
<UParam>	int	UParam value
<VParam>	int	UParam value
<XPosition>	string	X position value
<YPosition>	string	Y position value
<ZPosition>	string	Z position value
<WhichAxis>	string	X, Y, or Z
<ZRotationAngle>	string	0deg, 90deg, 180deg, or 270deg
<XOffset>	string	X offset value
<YOffset>	string	Y offset value
<AutoAxis>	bool	False
<SubmodelDefinitionName>	string	Component name
<NAME>	string	ComponentPriorityLists
<NextUniqueID>	int	Unique identification number
<MoveBackwards>	bool	False
<DatasetType>	string	ComponentDatasetType
<NAME>	string	DatasetDefinitions
<NAME>	string	BasicComponentInfo
<ComponentName>	string	Component name
<Company>	string	Component company name
<Company URL>	string	Component company website address
<Model Number>	string	Component model number
<Help URL>	string	Component online help website address
<Version>	list	Component version
<Notes>	string	Component notes
<IconType>	string	CTM
<NAME>	string	GeometryDefinitionParameters
<NAME>	string	VariableOrders
<NAME>	string	DesignDefinitionParameters

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	VariableOrders
	<NAME>	string	MaterialDefinitionParameters
	<NAME>	string	VariableOrders
	<MapInstanceParameters>	string	DesignVariable
	<UniqueDefinitionIdentifier>	string	Unique ID number
	<OriginFilePath>	string	Component file path
	<IsLocal>	bool	True or False
	<ChecksumString>	string	Checksum
	<ChecksumHistory>	list	Checksum history
	<VersionHistory>	list	Version history
	<NAME>	string	NativeComponentDefinitionProvider
	<Type>	string	CTM
	<Unit>	string	Unit type
	<MovePlane>	string	XY, YZ, or XZ
	<CTM import file>	int	File path to CTM powermap file
	<Rotation>	string	0deg, 90deg, 180deg, or 270deg
	<Direction>	string	Up, Down(X), or Down(Y)
	<Density number>	string	Density number value
	<Resolution>	string	Resolution value
	<Radiation>	string	True or False
	<RadiationSurfaceMaterial>	string	Surface material name
	<NAME>	string	InstanceParameters
	<GeometryParameters>	string	Geometry parameters list
	<MaterialParameters>	string	Material parameters list
	<DesignParameters>	string	Design parameters list
Return Value	None		

Python Syntax	InsertNativeComponent (*<NAME>*, *<NAME>*, *<KernelVersion>*, *<PartID>*, *<NAME>*, *<IsAttachedToEntity>*, *<EntityID>*, *<FacetedBodyTriangleIndex>*, *<TriangleVertexIndex>*, *<PositionType>*, *<UParam>*, *<VParam>*, *<XPosition>*, *<YPosition>*, *<ZPosition>*, *<MoveToEnd>*, *<FaceID>*, *<NAME>*, *<IsAttachedToEntity>*, *<EntityID>*, *<FacetedBodyTriangleIndex>*, *<TriangleVertexIndex>*, *<PositionType>*, *<UParam>*, *<VParam>*, *<XPosition>*, *<YPosition>*, *<ZPosition>*, *<WhichAxis>*, *<ZRotationAngle>*, *<XOffset>*, *<YOffset>*, *<AutoAxis>*, *<SubmodelDefinitionName>*, *<NAME>*, *<NextUniqueID>*, *<MoveBackwards>*, *<DatasetType>*, *<NAME>*, *<NAME>*, *<ComponentName>*, *<Company>*, *<Company URL>*, *<Model Number>*, *<Help URL>*, *<Version>*, *<Notes>*, *<IconType>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<MapInstanceParameters>*, *<UniqueDefinitionIdentifier>*, *<OriginFilePath>*, *<IsLocal>*, *<ChecksumString>*, *<ChecksumHistory>*, *<VersionHistory>*, *<NAME>*, *<Type>*, *<Unit>*, *<MovePlane>*, *<CTM import file>*, *<Rotation>*, *<Density number>*, *<Res-*

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

	olution>, <*Radiation*>, <*RadiationSurfaceMaterial*>, <*NAME*>, <*Geometry-Parameters*>, <*MaterialParameters*>, <*DesignParameters*>)
Python Example	oEditor.InsertNativeComponent(["NAME:InsertNativeComponentData", ["NAME:FaceCSParameters", "KernelVersion:=", 13, "PartID:=", 34, ["NAME:Origin", "IsAttachedToEntity:=", True, "EntityID:=", 35, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=", -1, "PositionType:=", "FaceCenter", "UParam:=", 0, "VParam:=", 0, "XPosition:=", "0", "YPosition:=", "0", "ZPosition:=", "0"], "MoveToEnd:=", False, "FaceID:=", 35, ["NAME:AxisPosn", "IsAttachedToEntity:=", True, "EntityID:=", 41, "FacetedBodyTriangleIndex:=", -1, "TriangleVertexIndex:=", -1, "PositionType:=", "EdgeCenter", "UParam:=", 0, "VParam:=", 0, "XPosition:=", "0", "YPosition:=", "0", "ZPosition:=", "0"], "WhichAxis:=", "X", "ZRotationAngle:=", "0deg", "XOffset:=", "0", "YOffset:=", "0", "AutoAxis:=", False], "SubmodelDefinitionName:=", "CTM1",

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

```
                [
                        "NAME:ComponentPriorityLists"
                ],
                "NextUniqueID:="        , 0,
                "MoveBackwards:="       , False,
                "DatasetType:="         , "ComponentDatasetType",
                [
                        "NAME:DatasetDefinitions"
                ],
                [
                        "NAME:BasicComponentInfo",
                        "ComponentName:="       , "CTM1",
                        "Company:="             , "",
                        "Company URL:="         , "",
                        "Model Number:="        , "",
                        "Help URL:="            , "",
                        "Version:="             , "1.0",
                        "Notes:="               , "",
                        "IconType:="            , "CTM"
                ],
                [
                        "NAME:GeometryDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:DesignDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:MaterialDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                "MapInstanceParameters:=", "NotVariable",
                "UniqueDefinitionIdentifier:=", "490f7694-de68-4200-8e6b-
eefaf4f3d4d4",
                "OriginFilePath:="      , "",
                "IsLocal:="             , False,
                "ChecksumString:="      , "",
                "ChecksumHistory:="     , [],
                "VersionHistory:="      , [],
                [
                        "NAME:NativeComponentDefinitionProvider",
```


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

InsertNativeComponent

Creates a heatsink component in an Icepak design.

UI Access	**Project Manager** > **3D Component** > **Create** > **Package**		
Parameters	Name	Type	Description
	<NAME>	string	Package name
	<TargetCS>	string	Coordinate system that the package exists on
	<SubmodelDefinitionName>	string	PBGA, CavityDownBGA, FPBGA, Flip-chip BGA, QFP, QFN, or Dual
	<NAME>	string	ComponentPriorityLists
	<NextUniqueID>	int	0
	<MoveBackwards>	bool	True or False
	<DatasetType>	string	ComponentDatasetType
	<NAME>	string	DatasetDefinitions
	<NAME>	string	BasicComponentInfo
	<ComponentName>	string	Name of package component
	<Company>	string	Name of manufacturer
	<Company URL>	string	Manufacturer website URL
	<Model Number>	string	Manufacturer model number
	<Help URL>	string	Manufacturer online help URL
	<Version>	int	Package component version number
	<IsCompV2>	bool	True or False
	<RepGrpID>	int	Component group ID number
	<Notes>	string	Notes on packagecomponent
	<IconType>	string	Package
	<NAME>	string	GeometryDefinitionParameters
	<NAME>	string	GeometryDefinitionParameters
	<NAME>	string	VariableOrders
	<NAME>	string	DesignDefinitionParameters
	<NAME>	string	VariableOrders
	<NAME>	string	MaterialDefinitionParameters
	<DefReferenceCSID>	int	VariableOrders
	<MapInstanceParameters>	string	DesignVariable
	<UniqueDefinitionIdentifier>	string	ID number
	<OriginFilePath>	string	File path
	<CompV2GUID>	int	Component v2 ID number
	<CompAcmpFile>	string	Component acmp 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.

<IsCompV2>	bool	True or False
<RepGrpID>	int	Group ID number
<IsLocal>	bool	True or False
<ChecksumString>	string	Checksum string text
<ChecksumHistory>	list	Checksum history text
<VersionHistory>	list	Version history text
<NAME>	string	NativeComponentDefinitionProvider
<Type>	string	Package
<Unit>	string	Length unit
<Version>	int	Component version number
<PackageType>	string	PBGA, CavityDownBGA, FPBGA, Flip-chip BGA, QFP, QFN, or Dual
<ModelType>	string	Detailed, Compact Conduction Model, Characterization JC, or Characterization JB
<SymmetryType>	string	Full, Half, or Quarter
<ArrayType>	string	Full Array or Peripheral Array
<EnableCentralBalls>	bool	True or False
<BallShapeType>	string	Cylinder or Block
<DieSourceType>	string	2D Source
<EnableChamferedDiePaddle>	bool	True of False
<EnableHeatsink>	bool	True or False
<BumpModelType>	string	Simple
<LeadDirection>	string	Along Length
<Materials>	list	Material assignments for package components
<GeomToMaterialMap>	list	
<MovePlane>	string	Package plan
<XStart>	string	X starting position
<YStart>	string	Y starting position
<ZStart>	string	Y starting position
<Length>	string	Package length and unit
<Width>	string	Package width and unit
<PackageThickness>	string	Package thickness and unit
<NumberOfRowsAlongLength>	string	Number of rows on package length
<NumberOfRowsAlongWidth>	string	Number of rows on package width
<BallPitch>	string	Solder ball pitch and unit
<BallDiameter>	string	Solder ball diameter and unit
<BallHeight>	string	Solder ball height and unit
<MaskThickness>	string	Solder mask thickness and unit

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

<NumberOfSubstrateLayers>	string	Number of substrate layers in package
<SubstrateThickness>	string	Substrate thickness and unit
<TopTraceCoverage>	string	Trace coverage percentage on top of package
<BottomTraceCoverage>	string	Trace coverage percentage on bottom of package
<FirstIntLayerConverage>	string	First internal layer trace coverage
<SecondIntLayerConverage>	string	Second internal layer trace coverage
<TraceThickness>	string	Trace thickness and unit
<NumberOfThermalVias>	string	Number of vias in package
<ViaDiameter>	string	Via diameter and unit
<ViaPlateThickness>	string	Via plate thickness and unit
<DieLength>	string	Die length and unit
<DieWidth>	string	Die width and unit
<DieHeight>	string	Die height and unit
<DiePower>	string	Die power and unit
<DiePaddleLength>	string	Die paddle length and unit
<DiePaddleWidth>	string	Die paddle width and unit
<DiePaddleThickness>	string	Die paddle thickness and unit
<DieAttachThickness>	string	Die attach thickness and unit
<WireBondDiameter>	string	Wirebond diameter and unit
<WireBondTotalLength>	string	Wirebond total length and unit
<Enable top side condition>	bool	True or False
<Top Side Condition>	string	Heat Transfer Coefficient
<Enable top surface radiation>	bool	True or False
<Enable bottom side condition>	bool	True or False
<Bottom side condition>	string	Heat Transfer Coefficient
<Enable bottom side condition>	bool	True or False
<HeatFluxTop>	string	Heat flux value and unit for top side of package
<TemperatureTop>	string	AmbientTemp or temperature value and unit for top side of package
<HTCTop>	string	Heat transfer coefficient for top side of package
<RefTemperatureTop>	string	AmbientTemp or temperature value and unit for top side of package
<HeatFluxBottom>	string	Heat flux value and unit for bottom side of package
<TemperatureBottom>	string	AmbientTemp or temperature value and unit for bottom side of package

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

	<HTCBottom>	string	Heat transfer coefficient for bottom side of package
	<RefTemperatureBottom>	string	AmbientTemp or temperature value and unit for bottom side of package
	<NAME>	string	InstanceParameters
	<GeometryParameters>	string	Heatsink geometry parameter values
	<MaterialParameters>	string	Heatsink material parameter values
	<DesignParameters>	string	Heatsink design parameter values
Return Value	None		

InsertNativeComponent (*<NAME>*, *<TargetCS>*, *<SubmodelDefinitionName>*, *<NAME>*, *<NextUniqueID>*, *<MoveBackwards>*, *<DatasetType>*, *<NAME>*, *<NAME>*, *<ComponentName>*, *<Company>*, *<Company URL>*, *<Model Number>*, *<Help URL>*, *<Version>*, *<IsCompV2>*, *<RepGrpID>*, *<Notes>*, *<IconType>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<NAME>*, *<DefReferenceCSID>*, *<MapInstanceParameters>*, *<UniqueDefinitionIdentifier>*, *<OriginFilePath>*, *<CompV2GUID>*, *<CompAcmpFile>*, *<IsCompV2>*, *<RepGrpID>*, *<IsLocal>*, *<ChecksumString>*, *<ChecksumHistory>*, *<VersionHistory>*, *<NAME>*, *<Type>*, *<Unit>*, *<Version>*, *<PackageType>*, *<ModelType>*, *<SymmetryType>*, *<ArrayType>*, *<EnableCentralBalls>*, *<BallShapeType>*, *<DieSourceType>*, *<EnableChamferedDiePaddle>*, *<EnableHeatsink>*, *<BumpModelType>*, *<LeadDirection>*, *<Materials>*, *<GeomToMaterialMap>*, *<MovePlane>*, *<XStart>*, *<YStart>*, *<ZStart>*, *<Length>*, *<Width>*, *<PackageThickness>*, *<NumberOfRowsAlongLength>*, *<NumberOfRowsAlongWidth>*, *<BallPitch>*, *<BallDiameter>*, *<BallHeight>*, *<MaskThickness>*, *<NumberOfSubstrateLayers>*, *<SubstrateThickness>*, *<TopTraceCoverage>*, *<BottomTraceCoverage>*, *<FirstIntLayerConverage>*, *<SecondIntLayerConverage>*, *<TraceThickness>*, *<NumberOfThermalVias>*, *<ViaDiameter>*, *<ViaPlateThickness>*, *<DieLength>*, *<DieWidth>*, *<DieHeight>*, *<DiePower>*, *<DiePaddleLength>*, *<DiePaddleWidth>*, *<DiePaddleThickness>*, *<DieAttachThickness>*, *<WireBondDiameter>*, *<WireBondTotalLength>*, *<Enable top side condition>*, *<Top Side Condition>*, *<Enable top surface radiation>*, *<Enable bottom side condition>*, *<Bottom side condition>*, *<Enable bottom side condition>*, *<HeatFluxTop>*, *<TemperatureTop>*, *<HTCTop>*, *<RefTemperatureTop>*, *<HeatFluxBottom>*, *<TemperatureBottom>*, *<HTCBottom>*, *<RefTemperatureBottom>*, *<NAME>*, *<GeometryParameters>*, *<MaterialParameters>*, *<DesignParameters>*)

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.InsertNativeComponent(
        [
                "NAME:InsertNativeComponentData",
                "TargetCS:="            , "Global",
                "SubmodelDefinitionName:=", "PBGA",
                [
                        "NAME:ComponentPriorityLists"
                ],
                "NextUniqueID:="        , 0,
                "MoveBackwards:="       , False,
                "DatasetType:="         , "ComponentDatasetType",
                [
                        "NAME:DatasetDefinitions"
                ],
                [
                        "NAME:BasicComponentInfo",
                        "ComponentName:="       , "PBGA",
                        "Company:="             , "",
                        "Company URL:="         , "",
                        "Model Number:="        , "",
                        "Help URL:="            , "",
                        "Version:="             , "1.0",
                        "IsCompV2:="            , False,
                        "RepGrpID:="            , -1,
                        "Notes:="               , "",
                        "IconType:="            , "Package"
                ],
                [
                        "NAME:GeometryDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:DesignDefinitionParameters",
                        [
                                "NAME:VariableOrders"
                        ]
                ],
                [
                        "NAME:MaterialDefinitionParameters",
                        [
                                "NAME:VariableOrders"
```


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

```
                        ]
                ],
                "DefReferenceCSID:="    , 1,
                "MapInstanceParameters:=", "DesignVariable",
                "UniqueDefinitionIdentifier:=", "0850487f-8e06-4f8c-ac88-
3b09d0f1b2c0",
                "OriginFilePath:="      , "",
                "CompV2GUID:="          , "",
                "CompAcmpFile:="        , "",
                "IsCompV2:="            , False,
                "RepGrpID:="            , -1,
                "IsLocal:="             , False,
                "ChecksumString:="      , "",
                "ChecksumHistory:="     , [],
                "VersionHistory:="      , [],
                [
                        "NAME:NativeComponentDefinitionProvider",
                        "Type:="                , "Package",
                        "Unit:="                , "mm",
                        "Version:="             , 0,
                        "PackageType:="         , "PBGA",
                        "ModelType:="           , "Detailed",
                        "SymmetryType:="        , "Full",
                        "ArrayType:="           , "Full Array",
                        "EnableCentralBalls:="  , False,
                        "BallShapeType:="       , "Block",
                        "DieSourceType:="       , "2D Source",
                        "EnableChamferedDiePaddle:=", True,
                        "EnableHeatsink:="      , False,
                        "BumpModelType:="       , "Simple",
                        "LeadDirection:="       , "Along Length",
                        "Materials:="           , [                         '
strateMaterial:="       , "Epoxy Resin-Typical",
pressedBallMaterial:=", "Air-solid",                                "BallMaterial:='
pb50_sn50",                             "MaskMaterial:="        , "Epoxy Resin-T
"DiePaddleMaterial:="   , "Si-Typical",                         "DieMaterial:="
ical",                          "HeatSinkMaterial:="    , "Si-Typical",
tachMaterial:=" , "Die_attach_material",                                    "WallMat
"Die_attach_material",                          "MoldMaterial:="        , "Mold_
"SolderBeneathMaterial:=", "Flipchip_underfill",
terial:="       , "Au-Typical",                         "BoardMaterial:="
"LeadMaterial:="        , "copper",                             "ThermalLandBene
"Cu-Pure",                              "DieUnderfillMaterial:=", "Solder-pb50_s
"HeatSpreaderMaterial:=", "Cu-Pure",                            "DieMaterial_2:=
solid"],
                        "GeomToMaterialMap:="   , [],
```


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

```
                "DiePaddleLength:="     , "11.9mm",
                        "DiePaddleWidth:="      , "11.9mm",
                        "DiePaddleThickness:="  , "0.1mm",
                        "DieAttachThickness:="  , "0.04mm",
                        "WireBondDiameter:="    , "0.025mm",
                        "WireBondTotalLength:=" , "1mm",
                        "Enable top side condition:=", False,
                        "Top Side Condition:="  , "Heat Transfer Coefficient",
                        "Enable top surface radiation:=", False,
                        "Enable bottom side condition:=", False,
                        "Bottom side condition:=", "Heat Transfer Coefficient",
                        "Enable bottom side condition:=", False,
                        "HeatFluxTop:="         , "1irrad_W_per_m2",
                        "TemperatureTop:="      , "AmbientTemp",
                        "HTCTop:="              , "10w_per_m2kel",
                        "RefTemperatureTop:="   , "AmbientTemp",
                        "HeatFluxBottom:="      , "1irrad_W_per_m2",
                        "TemperatureBottom:="   , "AmbientTemp",
                        "HTCBottom:="           , "10w_per_m2kel",
                        "RefTemperatureBottom:=", "AmbientTemp"
                ],
                [
                        "NAME:InstanceParameters",
                        "GeometryParameters:="  , "",
                        "MaterialParameters:="  , "",
                        "DesignParameters:="    , ""
                ]
        ])
```


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

13 - Monitor Script Commands

For *Icepak* designs, **Face Monitors** and **Point Monitors** are available:

```
oModule = oDesign.GetModule("Monitor")
```

Monitors are used for capturing and plotting flow and thermal data on faces and points assigned prior to solving the analysis setup. The following scripting commands are available:

AssignFaceMonitor

AssignPointMonitor

AssignFaceMonitor

Assigns a face monitor to selected face.

UI Access	**Icepak** > **Monitor** > **Assign** > **Face**		
Parameters	Name	Type	Description
	<NAME>	string	Please type in description manually
	<Quantities>	list	"MassFlow", "VolumeFlow", "Temperature", "TemperatureMaximum", "TemperatureMinimum", or "HeatFlowRate"
	<DefaultNameOver-ridden>	string	True or False
	<Faces>	list	Faces used to define monitor faces
Return Value	None		

Python Syntax	AssignFaceMonitor (*<NAME>*, *<Quantities>*, *<DefaultNameOverridden>*, *<Faces>*)
Python Example	`oModule.AssignFaceMonitor(`

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

```
        [
                "NAME:CPU_Face425",
                "Quantities:="          , ["TemperatureMaximum"],
                "DefaultNameOverridden:=", False,
                "Faces:="               , [425]
        ])
```

AssignPointMonitor

Assigns a point monitor to selected object. Point monitors are defined according to the type of entity specified (at a vertex, midpoint of an edge, centroid of a flat face, or centroid of an object).

UI Access	**Icepak** > **Monitor** > **Assign** > **Point**
Parameters	

Name	Type	Description
<NAME>	string	Point monitor name
<Quantities>	list	"Speed", "Pressure", "TKE", "Epsilon"," ViscosityRatio", "WallYPlus", "Temperature", "K_X", "K_Y", "K_Z", or "HeatFlux"
<DefaultNameOver-ridden>	string	True or False
<Objects>	list	Objects used to define monitor points

Return Value	None
Python Syntax	AssignPointMonitor (*<NAME>*, *<Quantities>*, *<DefaultNameOverridden>*, *<Objects>*)
Python Example	

```
oModule.AssignPointMonitor(
        [
                "NAME:CPU",
                "Quantities:="          , ["Temperature"],
```


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

	"DefaultNameOverridden:=", False, "Objects:=" , ["CPU"]])

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.

f

14 - Mesh Region Module Script Commands

Commands for mesh setup and operations should be executed by the "MeshSetup" module.

```
oModule = oDesign.GetModule("MeshRegion")
```

The topics for this section include:

Script Commands for Creating and Modifying Mesh Regions

General Commands Recognized by the Mesh Operations Module

Script Commands for Creating and Modifying Mesh Regions

Script commands for creating and modifying mesh operations are as follows:

EditGlobalMeshRegion

AssignVirtualMeshRegion

EditMeshRegion

EditGlobalMeshRegion

Defines Icepak global mesh region settings.

UI Access	**Icepak** > **Mesh** > **Edit Global Region**		
Parameters	Name	Type	Description
	<NAME>	string	Settings
	<MeshMethod>	string	MesherHD
	<UserSpecifiedSettings>	bool	"True" to enable advanced mesh settings or "False" to disable them
	<ComputeGap>	bool	"True" to enable minimum gap override settings or "False" to disable it
	Auto Mesh Setting		
	<MeshRegionResolution>	int	1, 2, 3, 4, or 5
	<GeometryBasedMeshing>	bool	"True" to enable mesh fusion of "False" to disable it
	<MinGapX>	string	Minimum gap X value and unit
	<MinGapY>	string	Minimum gap Y value and unit
	<MinGapZ>	string	Minimum gap Z value and unit

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><Objects></td><td>list</td><td>Region</td></tr><tr><td colspan="3">Advanced Mesh Settings</td></tr><tr><td><MaxElementSizeX></td><td>string</td><td>Maximum Element Size X value and unit</td></tr><tr><td><MaxElementSizeY></td><td>string</td><td>Maximum Element Size Y value and unit</td></tr><tr><td><MaxElementSizeZ></td><td>string</td><td>Maximum Element Size Z value and unit</td></tr><tr><td><MinElementsInGap></td><td>string</td><td>Minimum</td></tr><tr><td><MinElementsOnEdge></td><td>string</td><td>Minimum</td></tr><tr><td><MaxSizeRatio></td><td>string</td><td>Maximum</td></tr><tr><td><NoOGrids></td><td>bool</td><td>"True" to enable no O grid meshing or "False" to disable it</td></tr><tr><td><EnableMLM></td><td>bool</td><td>"True" to enable multi-level meshing or "False" to disable it</td></tr><tr><td><EnforeMLMType></td><td>string</td><td>3D or 2D</td></tr><tr><td><MaxLevels></td><td>string</td><td>Maximum levels value</td></tr><tr><td><BufferLayers></td><td>string</td><td>Buffer layers value</td></tr><tr><td><UniformMeshParametersType></td><td>string</td><td>Average or XYZ Max Sizes</td></tr><tr><td><StairStepMeshing></td><td>bool</td><td>"True" to enable stairstep meshing or "False" to disable it</td></tr><tr><td><MinGapX></td><td>string</td><td>Minimum gap X value and unit</td></tr><tr><td><MinGapY></td><td>string</td><td>Minimum gap Y value and unit</td></tr><tr><td><MinGapZ></td><td>string</td><td>Minimum gap Z value and unit</td></tr><tr><td><Objects></td><td>list</td><td>Region</td></tr></table>
Return Value	None

Python Syntax	EditGlobalMeshRegion (<*NAME*>, <*MeshMethod*>, <*UserSpecifiedSettings*>, <*ComputeGap*>, <*MeshRegionResolution*>, <*MinGapX*>, <*MinGapY*>, <*MinGapZ*>, <*Objects*>)

Python Example	`oModule.EditGlobalMeshRegion(`
	`[`
	`"NAME:Settings",`
	`"MeshMethod:=" , "MesherHD",`
	`"UserSpecifiedSettings:=", True,`
	`"ComputeGap:=" , True,`
	`"MaxElementSizeX:=" , "0.02667mm",`
	`"MaxElementSizeY:=" , "0.02667mm",`
	`"MaxElementSizeZ:=" , "0.02667mm",`
	`"MinElementsInGap:=" , "3",`
	`"MinElementsOnEdge:=" , "2",`

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

```
                "MaxSizeRatio:="        , "2",
                "NoOGrids:="            , False,
                "EnableMLM:="           , True,
                "EnforeMLMType:="       , "3D",
                "MaxLevels:="           , "0",
                "BufferLayers:="        , "0",
                "UniformMeshParametersType:=", "Average",
                "StairStepMeshing:="    , False,
                "MinGapX:="             , "1mm",
                "MinGapY:="             , "1mm",
                "MinGapZ:="             , "1mm",
                "Objects:="             , ["Region"]
                ])
```

AssignMeshRegion

Assigns a mesh region in an Icepak design.

UI Access: Right-click> **Assign Mesh Region**

Parameters

Name	Type	Description
<NAME>	string	Mesh region name
<Enable>	bool	"True" to enable the mesh region or "False" to disable it
<MeshMethod>	string	MesherHD
<UserSpecifiedSettings>	bool	"True" to enable advanced mesh settings or "False" to disable them
<MeshRegionResolution>	int	1, 2, 3, 4, or 5
<GeometryBasedMeshing>	bool	"True" to enable mesh fusion of "False" to disable it
<MinGapX>	string	Minimum gap X value and unit
<MinGapY>	string	Minimum gap Y value and unit
<MinGapZ>	string	Minimum gap Z value and unit
<Objects>	list	List of objects assigned to the mesh region
<ProximitySizeFunction>	bool	True or False
<CurvatureSizeFunction>	bool	True or False
<EnableTransition>	bool	True or False
<OptimizePCBMesh>	bool	True or False
<Enable2DCutCell>	bool	True or False
<EnforceCutCellMeshing>	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.

	<Enforce2dot5DCutCell>	bool	True or False
	<NAME>	string	Geometrical Attributes
	<MinSlackX>	string	-X padding value and unit
	<MaxSlackX>	string	+X padding value and unit
	<MinSlackY>	string	-Y padding value and unit
	<MaxSlackY>	string	+Y padding value and unit
	<MinSlackZ>	string	-Z padding value and unit
	<MaxSlackZ>	string	+Z padding value and unit
	<MinBboxX>	string	-X bounding box value and unit
	<MaxBboxX>	string	+X bounding box value and unit
	<MinBboxY>	string	-Y bounding box value and unit
	<MaxBboxY>	string	+Y bounding box value and unit
	<MinBboxZ>	string	-Z bounding box value and unit
	<MaxBboxZ>	string	+Z bounding box value and unit
Return Value	None		

Python Syntax	AssignMeshRegion (*<NAME>*, *<Enable>*, *<MeshMethod>*, *<UserSpecifiedSettings>*, *<MeshRegionResolution>*, *<GeometryBasedMeshing>*, *<MinGapX>*, *<MinGapY>*, *<MinGapZ>*, *<Objects>*, *<ProximitySizeFunction>*, *<CurvatureSizeFunction>*, *<EnableTransition>*, *<OptimizePCBMesh>*, *<Enable2DCutCell>*, *<EnforceCutCellMeshing>*, *<Enforce2dot5DCutCell>*, *<NAME>*, *<MinSlackX>*, *<MaxSlackX>*, *<MinSlackY>*, *<MaxSlackY>*, *<MinSlackZ>*, *<MaxSlackZ>*, *<MinBboxX>*, *<MaxBboxX>*, *<MinBboxY>*, *<MaxBboxY>*, *<MinBboxZ>*, *<MaxBboxZ>*)

Python Example	`oModule.AssignMeshRegion(`
	`[`
	`"NAME:MeshRegion1",`
	`"Enable:=" , True,`
	`"MeshMethod:=" , "MesherHD",`
	`"UserSpecifiedSettings:=", False,`
	`"MeshRegionResolution:=", 3,`
	`"GeometryBasedMeshing:=", True,`
	`"MinGapX:=" , "1mm",`
	`"MinGapY:=" , "1mm",`
	`"MinGapZ:=" , "1mm",`
	`"Objects:=" , ["SubRegion"],`
	`"ProximitySizeFunction:=", True,`
	`"CurvatureSizeFunction:=", True,`
	`"EnableTransition:=" , False,`
	`"OptimizePCBMesh:=" , True,`
	`"Enable2DCutCell:=" , False,`
	`"EnforceCutCellMeshing:=", False,`

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

```
                "Enforce2dot5DCutCell:=", False
        ],
        [
                "NAME:Geometrical Attributes",
                "MinSlackX:="           , "0mm",
                "MaxSlackX:="           , "0mm",
                "MinSlackY:="           , "0mm",
                "MaxSlackY:="           , "0mm",
                "MinSlackZ:="           , "0mm",
                "MaxSlackZ:="           , "0mm",
                "MinBboxX:="            , "0mm",
                "MaxBboxX:="            , "0mm",
                "MinBboxY:="            , "0mm",
                "MaxBboxY:="            , "0mm",
                "MinBboxZ:="            , "0mm",
                "MaxBboxZ:="            , "0mm"
        ])
```

AssignMeshOperation

Assigns a mesh operation to model geometry.

UI Access	Right-click > **Assign Mesh Operation**

Parameters

Name	Type	Description
<NAME>	string	Mesh operation name
<Enable>	bool	"True" to enable the mesh region or "False" to disable it
<Local Mesh Parameters Enabled>	bool	"True" to enable the mesh region or "False" to disable it
<Local Mesh Parameters Type>	string	"Box Local Mesh Parameters", "Cylinder Local Mesh Parameters", "Circle Local Mesh Parameters", "Quad Local Mesh Parameters", "2DPolygon Local Mesh Parameters", "3DPolygon Local Mesh Parameters", "Face Local Mesh Parameters", or "Edge Local Mesh Parameters",
<Objects>	list	For objects: list of objects assigned to the mesh region
<Faces>		For faces: list of faces assigned to the mesh region
<Edges>		For edges: list of faces assigned to the mesh region
<MaxLevel>	string	Maximum number of mesh levels
<MinLevel>	string	Minimum number of mesh levels
<IncrLevel>	string	Positive or negative increment level
<Mesh Reuse Enabled>	bool	"True" to enable the mesh region or "False" to disable it

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

	<Mesh Reuse File>	string	File path to mesh reuse file
	<Mesh Object(s) Separately Enabled>	bool	"True" to enable the mesh region or "False" to disable it
Return Value	None		

Python Syntax	AssignMeshOperation (*<NAME>*, *<Enable>*, *<Local Mesh Parameters Enabled>*, *<Objects>*, *<MaxLevel>*, *<MinLevel>*, *<Mesh Reuse Enabled>*, *<Mesh Reuse File>*, *<Mesh Object(s) Separately Enabled>*)

Python Example	`oModule.AssignMeshOperation(`
	`[`
	`"NAME:MeshOperation1",`
	`"Enable:=" , True,`
	`"Local Mesh Parameters Enabled:=", True,`
	`"Local Mesh Parameters Type:=", "Box Local Mesh Parameters",`
	`"Objects:=" , ["Box2"],`
	`"X count:=" , "1",`
	`"Y count:=" , "1",`
	`"Z count:=" , "1",`
	`"Inward height:=" , "1mm",`
	`"Inward ratio:=" , "0.1",`
	`"MaxLevel:=" , "1",`
	`"MinLevel:=" , "1",`
	`"MaxLevel:=" , "0",`
	`"MinLevel:=" , "0",`
	`"Mesh Reuse Enabled:=" , False,`
	`"Mesh Reuse File:=" , "",`
	`"Mesh Object(s) Separately Enabled:=", False`
	`])`

AssignMeshOperation (Mesh Sizes Region)

Assigns a mesh size region to non-model geometry.

UI Access	Right-click > **Assign Mesh Operation** (non-model geometry)		
Parameters	Name	Type	Description
	<NAME>	string	Mesh size region name
	<Enable>	bool	"True" to enable the mesh region or "False" to disable it
	<Local Mesh Parameters Enabled>	bool	"True" to enable the mesh region or "False" to disable it
	<Local Mesh Parameters Type>	string	"Size region 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><*Objects*></td><td>list</td><td>List of objects assigned to the mesh region</td></tr><tr><td><*Element Size X*></td><td>string</td><td>X direction mesh element size value and unit</td></tr><tr><td><*Element Size Y*></td><td>string</td><td>Y direction mesh element size value and unit</td></tr><tr><td><*Element Size Z*></td><td>string</td><td>Z direction mesh element size value and unit</td></tr></table>
Return Value	None

Python Syntax	AssignMeshOperation (<*NAME*>, <*Enable*>, <*Local Mesh Parameters Enabled*>, <*Local Mesh Parameters Type*>, <*Objects*>, <*Element Size X*>, <*Element Size Y*>, <*Element Size Z*>)

Python Example	`oModule.AssignMeshOperation(`
	`  [`
	`  "NAME:SizeRegion1",`
	`  "Enable:="  , True,`
	`  "Local Mesh Parameters Enabled:=", True,`
	`  "Local Mesh Parameters Type:=", "Size region Parameters",`
	`  "Objects:="  , ["Box1"],`
	`  "Element Size X:="  , "0.1mm",`
	`  "Element Size Y:="  , "0.1mm",`
	`  "Element Size Z:="  , "0.1mm"`
	`  ])`

AssignVirtualMeshRegion

AssignVirtualMeshRegion allows you to assign a mesh region in Icepak.

UI Access	Right-click geometry > **Assign Mesh Region**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*NAME*></td><td>string</td><td>Mesh region name</td></tr><tr><td><*Enable*></td><td>bool</td><td>True or False</td></tr><tr><td><*MeshMethod*></td><td>string</td><td>MesherHD</td></tr><tr><td><*UserSpecifiedSettings*></td><td>bool</td><td>True or False</td></tr><tr><td><*MeshRegionResolution*></td><td>int</td><td>Auto mesh setting value (1-5)</td></tr><tr><td><*MinGapX*></td><td>string</td><td>Minimum gap X value and unit</td></tr><tr><td><*MinGapY*></td><td>string</td><td>Minimum gap Y value and unit</td></tr><tr><td><*MinGapZ*></td><td>string</td><td>Minimum gap Z value and unit</td></tr><tr><td><*Objects*></td><td>list</td><td>Selected objects</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.

	<ProximitySizeFunction>	bool	True or False
	<CurvatureSizeFunction>	bool	True or False
	<EnableTransition>	bool	True or False
	<OptimizePCBMesh>	bool	True or False
	<Enable2DCutCell>	bool	True or False
	<EnforceCutCellMeshing>	bool	True or False
	<Enforce2dot5DCutCell>	bool	True or False
	<SlackMinX>	string	Slack minimum X direction value and unit
	<SlackMinY>	string	Slack minimum Y direction value and unit
	<SlackMinZ>	string	Slack minimum Z direction value and unit
	<SlackMaxX>	string	Slack maximum X direction value and unit
	<SlackMaxY>	string	Slack maximum Y direction value and unit
	<SlackMaxZ>	string	Slack maximum Z direction value and unit
Return Value	None		

Python Syntax	AssignVirtualMeshRegion (*<NAME>*, *<Enable>*, *<MeshMethod>*, *<UserSpecifiedSettings>*, *<MeshRegionResolution>*, *<MinGapX>*, *<MinGapY>*, *<MinGapZ>*, *<Objects>*, *<ProximitySizeFunction>*, *<CurvatureSizeFunction>*, *<EnableTransition>*, *<OptimizePCBMesh>*, *<Enable2DCutCell>*, *<EnforceCutCellMeshing>*, *<Enforce2dot5DCutCell>*, *<SlackMinX>*, *<SlackMinY>*, *<SlackMinZ>*, *<SlackMaxX>*, *<SlackMaxY>*, *<SlackMaxZ>*, *<NAME>*, *<MinSlackX>*, *<MaxSlackX>*, *<MinSlackY>*, *<MaxSlackY>*, *<MinSlackZ>*, *<MaxSlackZ>*, *<MinBboxX>*, *<MaxBboxX>*, *<MinBboxY>*, *<MaxBboxY>*, *<MinBboxZ>*, *<MaxBboxZ>*)

Python Example	`oModule.AssignVirtualMeshRegion(`
	`[`
	`"NAME:MeshRegion1",`
	`"Enable:=" , True,`
	`"MeshMethod:=" , "MesherHD",`
	`"UserSpecifiedSettings:=", False,`
	`"MeshRegionResolution:=", 3,`
	`"MinGapX:=" , "1mm",`
	`"MinGapY:=" , "1mm",`
	`"MinGapZ:=" , "1mm",`
	`"Objects:=" , ["HEAT_SINK"],`
	`"ProximitySizeFunction:=", True,`
	`"CurvatureSizeFunction:=", True,`

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

```
                "EnableTransition:="    , False,
                "OptimizePCBMesh:="     , True,
                "Enable2DCutCell:="     , False,
                "EnforceCutCellMeshing:=", False,
                "Enforce2dot5DCutCell:=", False,
                "SlackMinX:="           , "5mm",
                "SlackMinY:="           , "5mm",
                "SlackMinZ:="           , "5mm",
                "SlackMaxX:="           , "5mm",
                "SlackMaxY:="           , "5mm",
                "SlackMaxZ:="           , "5mm"
        ],
```

EditMeshRegion

This command is used to modify an existing mesh region operation. It has been deprecated; use the Edit command instead.

The following is an example of editing a mesh region operation with the Edit command:

```
oModule = oDesign.GetModule("MeshSetup")
oModule.Edit("MeshRegion1",
  [
    "NAME:MeshRegion2",
    "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.

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.

15 - Analysis Setup Module Script Commands

Icepak analysis setup commands should be executed by the Analysis module, referred to in Icepak scripts as the "AnalysisSetup" module.

```
oModule = oDesign.GetModule("AnalysisSetup")
```

AddTwoWayCoupling

ClearLinkedData

CopySetup

CopySweep

DeleteSetups

DeleteTwoWayCoupling

EditSetup

EditTwoWayCoupling

GetSetupCount

GetSetups

InsertSetup (Icepak Steady State)

InsertSetup (Icepak Transient)

PasteSetup

RenameSetup

RevertAllToInitial

RevertSetupToInitial

RevertSetupToInitialCondition

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

ClearLinkedData (Module)

Clear the linked data of the specified solution setups. (This command is similar to the ClearLinkedData command for the *design* level, which clears the linked data for all solution setups in the design.)

UI Access	**Project Manager** > ***{Design name}*** > **Analysis** > right-click ***{Setup name}*** > **Clear Linked Data** or, with a solution setup selected in the Project Manager: **Icepak** > **Analysis** > **Clear Linked Data**
Parameters	Name / Type / Description <*SetupNameArray*> / Array / Specify the name of the setups whose linked data are to be cleaned
Return Value	None

Python Syntax	ClearLinkedData(<SetupNameArray>)
Python Example	`oModule.ClearLinkedData(["setup1"])`

CopySetup

Copy 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	CopySetup (<*SetupName*>)
Python Example	`oModule.CopySetup ("OptimizationSetup1")`

DeleteSetups

Deletes one or more solution setups, which are specified by an array of solution setup names.

UI Access	Right-click a solution setup in the project tree and then click **Delete** on the shortcut menu, or delete selected solution setups in the **List** dialog box.
Parameters	Name | Type | Description <*SetupArray*> | Array | Array of solution setup names.
Return Value	None.

Python Syntax	DeleteSetups (<*SetupArray*>)
Python Example	`oModule.DeleteSetups(["Setup1", "Setup2"])`

EditSetup

Modifies an existing 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.

<table>
<tr><td>UI Access</td><td>Double-click a solution setup in the project tree to modify its settings.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SetupName></td><td>String</td><td>Name of the solve setup being edited.</td></tr>
<tr><td><Attributes></td><td>Array</td><td>Structured array.
["NAME:<NewSetupName>", <NamedParameters>]
See the InsertSetup command for details and examples.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	EditSetup (<*SetupName*>, <*Attributes*>)
Python Example	oModule.EditSetup("Setup1", ["NAME:NewSetup", "AdaptiveFreq:=", "1GHz", "EnableDistribProbTypeOption:=", false, "SaveFields:=", "true", "Enabled:=", true, ["NAME:Cap", "MaxPass:=", 10, "MinPass:=", 1, "MinConvPass:=", 2,

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

```
        "PerError:=", 1,
        "PerRefine:=", 30,
        "AutoIncreaseSolutionOrder:=", false,
        "SolutionOrder:=", "Normal"],
["NAME:DC",
        "Residual:=", 1E-005,
        "SolveResOnly:=", false,
        ["NAME:Cond",
                "MaxPass:=", 10,
                "MinPass:=", 1,
                "MinConvPass:=", 1,
                "PerError:=", 1,
                "PerRefine:=", 30),
        ["NAME:Mult",
                "MaxPass:=", 1,
                "MinPass:=", 1,
                "MinConvPass:=", 1,
                "PerError:=", 1,
                "PerRefine:=", 30]],
["NAME:AC",
        "MaxPass:=", 10,
```


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

```
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30]]
)
oModule.EditSetup("HfssDrivenAuto",
["NAME:Setup1",
  "IsEnabled:=", True,
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
            "RangeType:=", "LinearStep",
            "RangeStart:=", "1GHz",
            "RangeEnd:=", "10GHz",
            "RangeStep:=", "1GHz"
            ]
  ],
  "SaveRadFieldsOnly:=", False,
  "SaveAnyFields:=", True,
  "Type:=", "Discrete"
```


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

```
  ])


oModule.EditSetup("AC Magnetic",
[
  "NAME:AC Magnetic",
  "Enabled:="             , True,
  [
    "NAME:MeshLink",
    "ImportMesh:="          , False
  ],
  "MaximumPasses:="       , 4,
  "MinimumPasses:="       , 2,
  "MinimumConvergedPasses:=", 1,
  "PercentRefinement:="   , 30,
  "SolveFieldOnly:="      , False,
  "PercentError:="        , 0.1,
  "SolveMatrixAtLast:="   , True,
  "UseNonLinearIterNum:=" , False,
  [
    "NAME:ExpressionCache",
    [
```


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

```
      "NAME:CacheItem",
      "Title:="               , "eddy_loss1",
      "Expression:="          , "eddy_loss",
      "Intrinsics:="          , "Phase=\'0deg\'",
      "ReportType:="          , "Fields",
      [
        "NAME:ExpressionContext"
      ]
    ]
  ],
  "UseCacheFor:="         , ["Pass"],
  "UseIterativeSolver:="  , False,
  "RelativeResidual:="    , 0.0001,
  "NonLinearResidual:="   , 0.0001,
  "SmoothBHCurve:="       , False,
  "Frequency:="           , "200Hz",
  "HasSweepSetup:="       , False,
  "UseHighOrderShapeFunc:=", False,
  "UseMuLink:="           , False,
  "LossAdaptiveCtrl:="    , "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.

```
])
oModule.EditSetup("HfssDriven",
["NAME:Setup3",
        "AdaptMultipleFreqs:=", False,
        "Frequency:=", "5GHz",
        "MaxDeltaS:=", 0.02,
        "PortsOnly:=", False,
        "UseMatrixConv:=", False,
        "MaximumPasses:=", 6,
        "MinimumPasses:=", 1,
        "MinimumConvergedPasses:=", 1,
        "PercentRefinement:=", 30,
        "IsEnabled:=", True,
        "BasisOrder:=", 1,
        "DoLambdaRefine:=", True,
        "DoMaterialLambda:=", True,
        "SetLambdaTarget:=", False,
        "Target:=", 0.3333,
        "UseMaxTetIncrease:=", False,
        "PortAccuracy:=", 2,
        "UseABCOnPort:=", False,
```


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

```
        "SetPortMinMaxTri:=", False,
        "UseDomains:=", True,
        "UseIterativeSolver:=", False,
        "IterativeResidual:=", 1E-06,
        "DDMSolverResidual:=", 0.0001,
        "EnhancedLowFreqAccuracy:=", True,
        "SaveRadFieldsOnly:=", False,
        "SaveAnyFields:=", True,
        "IESolverType:=", "Auto",
        "LambdaTargetForIESolver:=", 0.15,
        "UseDefaultLambdaTgtForIESolver:=", True,
        "SkipIERegionSolveDuringAdaptivePasses:=", True
        "RayDensityPerWavelength:=", 4,
        "MaxNumberOfBounces:="  , 5,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "SkipSBRSolveDuringAdaptivePasses:=", True,
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
])
```

Edit an SBR+ Setup with Fast Frequency Looping

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

```
oModule.EditSetup("HfssDriven",
        [
                "NAME:Setup1",
                "IsEnabled:="           , True,
                [
                        "NAME:MeshLink",
                        "ImportMesh:="          , False
                ],
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
                "RadiationSetup:="      , "",
                "PTDUTDSimulationSettings:=", "None",
                "FastFrequencyLooping:=", True,
                [
                        "NAME:Sweeps",
                        [
                                "NAME:Sweep",
                                "RangeType:="           , "LinearStep",
```


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

```
                                "RangeStart:="          , "1GHz",
                                "RangeEnd:="            , "10GHz",
                                "RangeStep:="           , "1GHz"
                            ]
                    ],
                    "ComputeFarFields:="    , True
                    "UseSBREnhancedRadiatedPowerCalculation:=", True,
                    "IsGOBlockageEnabled:=" , False,
                    "GOBlockageSurfaceSelfBlock:=", False
            ])
```

Edit and RF Discharge Setup for HFSS

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("coaxbend_discharge_r212")
oDesign = oProject.SetActiveDesign("HFSSDesign60degBendTeflon")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.EditSetup("RFDischarge1",
  [
    "NAME:RFDischarge1",
```


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

```
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
      "Soln:="                , "Setup1 : Sweep",
      [
        "NAME:Params",
        "bend_angle:="          , "bend_angle"
      ],
      "ForceSourceToSolve:="  , True,
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "SourceProduct",
      "ApplyMeshOp:="         , True
    ],
    [
      "NAME:Excitations",
      [
        "NAME:1:1",
```


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

```
        "Magnitude:="           , "1",
        "Phase:="               , "0deg"
      ],
      [
        "NAME:2:1",
        "Magnitude:="           , "0",
        "Phase:="               , "0deg"
      ]
    ],
    [
      "NAME:Frequencies",
      "10GHz"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Helium"
```


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

	])

GetSetupCount

Gets the number of analysis setups in a design.

UI Access	N/A
Parameters	None.
Return Value	Integer containing number of setups.

Python Syntax	GetSetupCount ()
Python Example	`oModule.GetSetupCount()`

GetSetups

Gets the names of analysis setups in a design.

UI Access	N/A
Parameters	None.
Return Value	Array of analysis setup names

Python Syntax	GetSetups ()

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.GetSetups()`

InsertSetup (Icepak - Steady State)

Creates a solution setup in an Icepak design with a steady state solution type.

UI Access	**Icepak** > **Analysis** > **Add Solution Setup**

Parameters

Name	Type	Description
<NAME>	string	Solution setup name
<Enabled>	bool	"True" to include setup in the analysis or "False" to exclude it
<Flow Regime>	string	Laminar or Turbulent
<Turbulent Model Eqn>	string	ZeroEquation, TwoEquation, RNG, EnhancedTwoEquation, SpalartAllmaras, EnhancedRNG, RealizableTwoEquation, EnhancedRealizableTwoEquation, or kOmegaSST
<Include Temperature>	bool	"True" to include temperature calculation or "False" to exclude it
<Include Flow>	bool	"True" to include the flow calculation or "False" to exclude it
<Include Gravity>	bool	"True" to include the effect of gravity or "False" to exclude it
<Solution Initialization - X Velocity>	string	X direction velocity initialization value and unit
<Solution Initialization - Y Velocity>	string	Y direction velocity initialization value and unit
<Solution Initialization - Z Velocity>	string	Z direction velocity initialization value and unit
<Solution Initialization - Temperature>	string	AmbientTemp or temperature value and unit
<Solution Initialization - Turbulent Kinetic Energy>	string	Turbulent kinetic energy initialization value and unit
<Solution Initialization - Turbulent Dissipation Rate>	string	Turbulent dissipation rate initialization value and unit
<Solution Initialization - Specific Dissipation Rate>	string	Specific dissipation rate initialization value and unit

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

<Convergence Criteria - Flow>	string	Flow convergence value
<Convergence Criteria - Energy>	string	Energy convergence value
<Convergence Criteria - Turbulent Kinetic Energy>	string	Turbulent kinetic energy convergence value
<Convergence Criteria - Turbulent Dissipation Rate>	string	Turbulent dissipation rate convergence value
<Convergence Criteria - Specific Dissipation Rate>	string	Specific dissipation rate convergence value
<Convergence Criteria - Discrete Ordinates>	string	Discrete ordinates convergence value
<Convergence Criteria - Joule Heating>	string	Joule heating convergence value
<IsEnabled>	bool	Not applicable to Icepak Solve Setup
<Radiation Model>	string	Off, Discrete Ordinates Model, or Ray Tracing Model
<Solar Radiation Model>	string	Solar Radiation Calculator
<Solar Enable Participating Solids>	bool	True or False
<Solar Radiation - Scattering Fraction>	string	Value between 0 and 1
<Solar Radiation - North X>	string	X direction value
<Solar Radiation - North Y>	string	Y direction value
<Solar Radiation - North Z>	string	Z direction value
<Solar Radiation - Day>	int	Date value
<Solar Radiation - Month>	int	Month value
<Solar Radiation - Hours>	int	Hour value
<Solar Radiation - Minutes>	int	Minute value
<Solar Radiation - GMT>	string	Time zone value offset from GMT
<Solar Radiation - Latitude>	string	Latitude value
<Solar Radiation - Latitude Direction>	string	North or South
<Solar Radiation - Longitude>	string	Longitude value
<Solar Radiation - Longitude Direction>	string	East or West
<Solar Radiation - Ground Reflectance>	string	Value between 0 and 1
<Solar Radiation - Sunshine Fraction>	string	Value between 0 and 1
<Flow Iteration Per Radiation Iteration>	string	Number of flow iterations per radiation iteration

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

<ThetaDivision>	string	Discrete Ordinates only: theta divisions value
<PhiDivision>	string	Discrete Ordinates only: Phi divisions value
<ThetaPixels>	string	Discrete Ordinates only: theta pixels value
<PhiPixels>	string	Discrete Ordinates only: phi pixels value
<Maximum Radiation Iteration>	string	Ray Tracing only: Maximum number of radiation iterations
<Faces Per Surface Cluster>	string	Ray Tracing only: faces per surface cluster value
<Resolution>	string	Ray Tracing only: resolution value
<Under-relaxation - Pressure>	string	Pressure under-relaxation value
<Under-relaxation - Momentum>	string	Momentum under-relaxation value
<Under-relaxation - Temperature>	string	Temperature under-relaxation value
<Under-relaxation - Turbulent Kinetic Energy>	string	Turbulent kinetic energy under-relaxation value
<Under-relaxation - Turbulent Dissipation Rate>	string	Turbulent dissipation rate under-relaxation value
<Under-relaxation - Specific Dissipation Rate>	string	Specific dissipation rate under-relaxation value
<Under-relaxation - Joule Heating>	string	Joule heating under-relaxation value
<Discretization Scheme - Pressure>	string	Standard, Second, Body Force, or PRESTO
<Discretization Scheme - Momentum>	string	First or Second
<Discretization Scheme - Temperature>	string	First or Second
<Secondary Gradient>	bool	"True" to enable secondary gradient or "False" to disable it
<Discretization Scheme - Turbulent Kinetic Energy>	string	First or Second
<Discretization Scheme - Turbulent Dissipation Rate>	string	First or Second
<Discretization Scheme - Specific Dissipation Rate>	string	First or Second
<Discretization Scheme - Discrete Ordinates>	string	First or Second
<Linear Solver Type - Pressure>	string	flex, V, W, or F

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

<Linear Solver Type - Momentum>	string	flex, V, W, or F
<Linear Solver Type - Temperature>	string	flex, V, W, or F
<Linear Solver Type - Turbulent Kinetic Energy>	string	flex, V, W, or F
<Linear Solver Type - Turbulent Dissipation Rate>	string	flex, V, W, or F
<Linear Solver Type - Specific Dissipation Rate>	string	flex, V, W, or F
<Linear Solver Type - Joule Heating>	string	flex, V, W, or F
<Linear Solver Termination Criterion - Pressure>	string	Pressure termination criterion value
<Linear Solver Termination Criterion - Momentum>	string	Momentum termination criterion value
<Linear Solver Termination Criterion - Temperature>	string	Temperature termination criterion value
<Linear Solver Termination Criterion - Turbulent Kinetic Energy>	string	Turbulent kinetic energy termination criterion value
<Linear Solver Termination Criterion - Turbulent Dissipation Rate>	string	Turbulent dissipation rate termination criterion value
<Linear Solver Termination Criterion - Specific Dissipation Rate>	string	Specific dissipation rate termination criterion value
<Linear Solver Termination Criterion - Joule Heating>	string	Joule heating termination criterion value
<Linear Solver Residual Reduction Tolerance - Pressure>	string	Pressure residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Momentum>	string	Momentum residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Temperature>	string	Temperature residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Turbulent Kinetic Energy>	string	Turbulent kinetic energy residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Turbulent Dissipation Rate>	string	Turbulent dissipation residual reduction tolerance value

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

	<Linear Solver Residual Reduction Tolerance - Specific Dissipation Rate>	string	Specific dissipation rate residual reduction tolerance value
	<Linear Solver Residual Reduction Tolerance - Joule Heating>	string	Joule heating residual reduction tolerance value
	<Linear Solver Stabilization - Pressure>	string	None or BCGSTAB
	<Linear Solver Stabilization - Temperature>	string	None or BCGSTAB
	<Linear Solver Stabilization - Joule Heating>	string	None or BCGSTAB
	<Coupled pressure-velocity formulation>	bool	True or False
	<Turn off auto-pairing for grid interface creation>	bool	True or False
	<2D Profile Interpolation Method>	string	Constant, Inverse Distance Weighted, or Least Squares
	<Frozen Flow Simulation>	bool	"False" to disable frozen flow, which is not available for steady-state simulations
	<Sequential Solve of Flow and Energy Equations>	bool	"True" to solve equations sequentially or "False" to solve simultaneously
	<Convergence Criteria - Max Iterations>	int	Maximum number of iterations
Return Value	None		

Python Syntax	InsertSetup (<*NAME*>, <*Enabled*>, <*Flow Regime*>, <*Turbulent Model Eqn*>, <*Include Temperature*>, <*Include Flow*>, <*Include Gravity*>, <*Solution Initialization - X Velocity*>, <*Solution Initialization - Y Velocity*>, <*Solution Initialization - Z Velocity*>, <*Solution Initialization - Temperature*>, <*Solution Initialization - Turbulent Kinetic Energy*>, <*Solution Initialization - Turbulent Dissipation Rate*>, <*Solution Initialization - Specific Dissipation Rate*>, <*Convergence Criteria - Flow*>, <*Convergence Criteria - Energy*>, <*Convergence Criteria - Turbulent Kinetic Energy*>, <*Convergence Criteria - Turbulent Dissipation Rate*>, <*Convergence Criteria - Specific Dissipation Rate*>, <*Convergence Criteria - Discrete Ordinates*>, <*IsEnabled*>, <*Radiation Model*>, <*Flow Iteration*

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

	Per Radiation Iteration>, <*ThetaDivision*>, <*PhiDivision*>, <*ThetaPixels*>, <*PhiPixels*>, <*Under-relaxation - Pressure*>, <*Under-relaxation - Momentum*>, <*Under-relaxation - Temperature*>, <*Under-relaxation - Turbulent Kinetic Energy*>, <*Under-relaxation - Turbulent Dissipation Rate*>, <*Under-relaxation - Specific Dissipation Rate*>, <*Discretization Scheme - Pressure*>, <*Discretization Scheme - Momentum*>, <*Discretization Scheme - Temperature*>, <*Secondary Gradient*>, <*Discretization Scheme - Turbulent Kinetic Energy*>, <*Discretization Scheme - Turbulent Dissipation Rate*>, <*Discretization Scheme - Specific Dissipation Rate*>, <*Discretization Scheme - Discrete Ordinates*>, <*Linear Solver Type - Pressure*>, <*Linear Solver Type - Momentum*>, <*Linear Solver Type - Temperature*>, <*Linear Solver Type - Turbulent Kinetic Energy*>, <*Linear Solver Type - Turbulent Dissipation Rate*>, <*Linear Solver Type - Specific Dissipation Rate*>, <*Linear Solver Termination Criterion - Pressure*>, <*Linear Solver Termination Criterion - Momentum*>, <*Linear Solver Termination Criterion - Temperature*>, <*Linear Solver Termination Criterion - Turbulent Kinetic Energy*>, <*Linear Solver Termination Criterion - Turbulent Dissipation Rate*>, <*Linear Solver Termination Criterion - Specific Dissipation Rate*>, <*Linear Solver Residual Reduction Tolerance - Pressure*>, <*Linear Solver Residual Reduction Tolerance - Momentum*>, <*Linear Solver Residual Reduction Tolerance - Temperature*>, <*Linear Solver Residual Reduction Tolerance - Turbulent Kinetic Energy*>, <*Linear Solver Residual Reduction Tolerance - Turbulent Dissipation Rate*>, <*Linear Solver Residual Reduction Tolerance - Specific Dissipation Rate*>, <*Linear Solver Stabilization - Pressure*>, <*Linear Solver Stabilization - Temperature*>, <*Frozen Flow Simulation*>, <*Sequential Solve of Flow and Energy Equations*>, <*Convergence Criteria - Max Iterations*>)

Python Example	`oModule.InsertSetup("IcepakSteadyState",`
	`  [`
	`  "NAME:Setup1",`
	`  "Enabled:=" , True,`
	`  "Flow Regime:=" , "Turbulent",`
	`  "Turbulent Model Eqn:=" , "ZeroEquation",`
	`  "Include Temperature:=" , True,`
	`  "Include Flow:=" , True,`
	`  "Include Gravity:=" , False,`
	`  "Solution Initialization - X Velocity:=", "-0.01m_per_sec",`
	`  "Solution Initialization - Y Velocity:=", "0m_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.

```
                "Solution Initialization - Z Velocity:=", "0m_per_sec",
                "Solution Initialization - Temperature:=", "AmbientTemp",
                "Solution Initialization - Turbulent Kinetic Energy:=", "1m2_per_s2",
                "Solution Initialization - Turbulent Dissipation Rate:=", "1m2_per_s3",
                "Solution Initialization - Specific Dissipation Rate:=", "1diss_per_s",
                "Convergence Criteria - Flow:=", "0.001",
                "Convergence Criteria - Energy:=", "1e-07",
                "Convergence Criteria - Turbulent Kinetic Energy:=", "0.001",
                "Convergence Criteria - Turbulent Dissipation Rate:=", "0.001",
                "Convergence Criteria - Specific Dissipation Rate:=", "0.001",
                "Convergence Criteria - Discrete Ordinates:=", "1e-06",
                "Convergence Criteria - Joule Heating:=", "1e-07",
                "IsEnabled:="           , False,
                "Radiation Model:="     , "Discrete Ordinates Model",
                "Solar Radiation Model:=", "Solar Radiation Calculator",
                "Solar Enable Participating Solids:=", False,
                "Solar Radiation - Scattering Fraction:=", "0",
                "Solar Radiation - North X:=", "0",
                "Solar Radiation - North Y:=", "0",
                "Solar Radiation - North Z:=", "1",
                "Solar Radiation - Day:=", 1,
                "Solar Radiation - Month:=", 1,
                "Solar Radiation - Hours:=", 0,
                "Solar Radiation - Minutes:=", 0,
                "Solar Radiation - GMT:=", "0",
                "Solar Radiation - Latitude:=", "0",
                "Solar Radiation - Latitude Direction:=", "East",
                "Solar Radiation - Longitude:=", "0",
                "Solar Radiation - Longitude Direction:=", "North",
```


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

```
"1",
                "Under-relaxation - Turbulent Kinetic Energy:=", "0.8",
                "Under-relaxation - Turbulent Dissipation Rate:=", "0.8",
                "Under-relaxation - Specific Dissipation Rate:=", "0.8",
                "Under-relaxation - Joule Heating:=", "1",
                "Discretization Scheme - Pressure:=", "Standard",
                "Discretization Scheme - Momentum:=", "First",
                "Discretization Scheme - Temperature:=", "First",
                "Secondary Gradient:="  , False,
                "Discretization Scheme - Turbulent Kinetic Energy:=", "First",
                "Discretization Scheme - Turbulent Dissipation Rate:=", "First",
                "Discretization Scheme - Specific Dissipation Rate:=", "First",
                "Discretization Scheme - Discrete Ordinates:=", "First",
                "Linear Solver Type - Pressure:=", "V",
                "Linear Solver Type - Momentum:=", "flex",
                "Linear Solver Type - Temperature:=", "F",
                "Linear Solver Type - Turbulent Kinetic Energy:=", "flex",
                "Linear Solver Type - Turbulent Dissipation Rate:=", "flex",
                "Linear Solver Type - Specific Dissipation Rate:=", "flex",
                "Linear Solver Type - Joule Heating:=", "flex",
                "Linear Solver Termination Criterion - Pressure:=", "0.1",
                "Linear Solver Termination Criterion - Momentum:=", "0.1",
                "Linear Solver Termination Criterion - Temperature:=", "0.1",
                "Linear Solver Termination Criterion -
                 Turbulent Kinetic Energy:=", "0.1",
                "Linear Solver Termination Criterion -
                 Turbulent Dissipation Rate:=", "0.1",
                "Linear Solver Termination Criterion -
                 Specific Dissipation Rate:=", "0.1",
                "Linear Solver Termination Criterion - Joule Heating:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Pressure:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Momentum:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Temperature:=", "0.1",
                "Linear Solver Residual Reduction Tolerance -
```


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

InsertSetup (Icepak - Transient)

Creates a solution setup in an Icepak design with a transient solution type.

UI Access	**Icepak** > **Analysis** > **Add Solution Setup**		
Parameters	Name	Type	Description
	<NAME>	string	Solution setup name
	<Enabled>	bool	"True" to include setup in the analysis or "False" to exclude it
	<Flow Regime>	string	Laminar or Turbulent
	<Turbulent Model Eqn>	string	ZeroEquation, TwoEquation, RNG, EnhancedTwoEquation, SpalartAllmaras, EnhancedRNG, RealizableTwoEquation, EnhancedRealizableTwoEquation, or kOmegaSST
	<Include Temperature>	bool	"True" to include temperature calculation or "False" to exclude it
	<Include Flow>	bool	"True" to include the flow calculation or "False" to exclude it
	<Include Gravity>	bool	"True" to include the effect of gravity or "False" to exclude it
	<Solution Initialization - X Velocity>	string	X direction velocity initialization value and unit
	<Solution Initialization - Y Velocity>	string	Y direction velocity initialization value and unit
	<Solution Initialization - Z Velocity>	string	Z direction velocity initialization value and unit
	<Solution Initialization - Temperature>	string	AmbientTemp or temperature value and unit
	<Solution Initialization - Turbulent Kinetic Energy>	string	Turbulent kinetic energy initialization value and unit

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

<Solution Initialization - Turbulent Dissipation Rate>	string	Turbulent dissipation rate initialization value and unit
<Solution Initialization - Specific Dissipation Rate>	string	Specific dissipation rate initialization value and unit
<Convergence Criteria - Flow>	string	Flow convergence value
<Convergence Criteria - Energy>	string	Energy convergence value
<Convergence Criteria - Turbulent Kinetic Energy>	string	Turbulent kinetic energy convergence value
<Convergence Criteria - Turbulent Dissipation Rate>	string	Turbulent dissipation rate convergence value
<Convergence Criteria - Specific Dissipation Rate>	string	Specific dissipation rate convergence value
<Convergence Criteria - Discrete Ordinates>	string	Discrete ordinates convergence value
<Convergence Criteria - Joule Heating>	string	Joule heating convergence value
<IsEnabled>	bool	Not applicable to Icepak Solve Setup
<Radiation Model>	string	Off, Discrete Ordinates Model, or Ray Tracing Model
<Solar Radiation Model>	string	Solar Radiation Calculator
<Solar Enable Participating Solids>	bool	True or False
<Solar Radiation - Scattering Fraction>	string	Value between 0 and 1
<Solar Radiation - North X>	string	X direction value
<Solar Radiation - North Y>	string	Y direction value
<Solar Radiation - North Z>	string	Z direction value
<Solar Radiation - Day>	int	Date value
<Solar Radiation - Month>	int	Month value
<Solar Radiation - Hours>	int	Hour value
<Solar Radiation - Minutes>	int	Minute value
<Solar Radiation - GMT>	string	Time zone value offset from GMT
<Solar Radiation - Latitude>	string	Latitude value
<Solar Radiation - Latitude Direction>	string	North or South
<Solar Radiation - Longitude>	string	Longitude value
<Solar Radiation - Longitude Direction>	string	East or West
<Solar Radiation - Ground Reflectance>	string	Value between 0 and 1

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

<Solar Radiation - Sunshine Fraction>	string	Value between 0 and 1
<Flow Iteration Per Radiation Iteration>	string	Number of flow iterations per radiation iteration
<ThetaDivision>	string	Discrete Ordinates only: theta divisions value
<PhiDivision>	string	Discrete Ordinates only: Phi divisions value
<ThetaPixels>	string	Discrete Ordinates only: theta pixels value
<PhiPixels>	string	Discrete Ordinates only: phi pixels value
<Maximum Radiation Iteration>	string	Ray Tracing only: Maximum number of radiation iterations
<Faces Per Surface Cluster>	string	Ray Tracing only: faces per surface cluster value
<Resolution>	string	Ray Tracing only: resolution value
<Under-relaxation - Pressure>	string	Pressure under-relaxation value
<Under-relaxation - Momentum>	string	Momentum under-relaxation value
<Under-relaxation - Temperature>	string	Temperature under-relaxation value
<Under-relaxation - Turbulent Kinetic Energy>	string	Turbulent kinetic energy under-relaxation value
<Under-relaxation - Turbulent Dissipation Rate>	string	Turbulent dissipation rate under-relaxation value
<Under-relaxation - Specific Dissipation Rate>	string	Specific dissipation rate under-relaxation value
<Under-relaxation - Joule Heating>	string	Joule heating under-relaxation value
<Discretization Scheme - Pressure>	string	Standard, Second, Body Force, or PRESTO
<Discretization Scheme - Momentum>	string	First or Second
<Discretization Scheme - Temperature>	string	First or Second
<Secondary Gradient>	bool	"True" to enable secondary gradient or "False" to disable it
<Discretization Scheme - Turbulent Kinetic Energy>	string	First or Second
<Discretization Scheme - Turbulent Dissipation Rate>	string	First or Second
<Discretization Scheme - Specific Dissipation Rate>	string	First or Second
<Discretization Scheme - Discrete Ordinates>	string	First or Second
<Linear Solver Type - Pressure>	string	flex, V, W, or F

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

<Linear Solver Type - Momentum>	string	flex, V, W, or F
<Linear Solver Type - Temperature>	string	flex, V, W, or F
<Linear Solver Type - Turbulent Kinetic Energy>	string	flex, V, W, or F
<Linear Solver Type - Turbulent Dissipation Rate>	string	flex, V, W, or F
<Linear Solver Type - Specific Dissipation Rate>	string	flex, V, W, or F
<Linear Solver Type - Joule Heating>	string	flex, V, W, or F
<Linear Solver Termination Criterion - Pressure>	string	Pressure termination criterion value
<Linear Solver Termination Criterion - Momentum>	string	Momentum termination criterion value
<Linear Solver Termination Criterion - Temperature>	string	Temperature termination criterion value
<Linear Solver Termination Criterion - Turbulent Kinetic Energy>	string	Turbulent kinetic energy termination criterion value
<Linear Solver Termination Criterion - Turbulent Dissipation Rate>	string	Turbulent dissipation rate termination criterion value
<Linear Solver Termination Criterion - Specific Dissipation Rate>	string	Specific dissipation rate termination criterion value
<Linear Solver Termination Criterion - Joule Heating>	string	Joule heating termination criterion value
<Linear Solver Residual Reduction Tolerance - Pressure>	string	Pressure residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Momentum>	string	Momentum residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Temperature>	string	Temperature residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Turbulent Kinetic Energy>	string	Turbulent kinetic energy residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Turbulent Dissipation Rate>	string	Turbulent dissipation residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Specific Dissipation Rate>	string	Specific dissipation rate residual reduction tolerance value
<Linear Solver Residual Reduction Tolerance - Joule Heating>	string	Joule heating residual reduction tolerance value

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

<Linear Solver Stabilization - Pressure>	string	None or BCGSTAB
<Linear Solver Stabilization - Temperature>	string	None or BCGSTAB
<Linear Solver Stabilization - Joule Heating>	string	None or BCGSTAB
<Coupled pressure-velocity formulation>	bool	True or False
<2D Profile Interpolation Method>	string	Constant, Inverse Distance Weighted, or Least Squares
<Turn off auto-pairing for grid interface creation>	bool	True or False
<Frozen Flow Simulation>	bool	"True" to enable frozen flow or "False" to disable it
<Start Time>	bool	Start time value and unit
<Stop Time>	string	Stop time value and unit
<Time Step>	string	Time step duration value and unit
<Iterations per Time Step>	int	Number of iterations per time step
<Import Start Time>	bool	"True" to enable import start time or "False" to disable it
<SaveFieldsType>	string	None or Every N Steps
<N Steps>	string	Number of time steps
<NAME>	string	SweepRanges
<NAME>	string	Subrange
<RangeType>	string	LinearStep
<RangeStart>	string	Range start time and unit
<RangeEnd>	string	Range end time and unit
<RangeStep>	string	Time step duration and unit
<NAME>	string	Time Step Variation Data
<Variation Type>	string	Transient
Time Step Variation = Linear		
<Variation Function>	string	Linear
<Variation Value>	list	Comma-separated values and units: (S0, a)
Time Step Variation = Square Wave		
<Variation Function>	string	Square Wave

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><Variation Value></td><td>list</td><td>Comma-separated values and units: (S0, Phase, On Time, Off Time, Off Value)</td></tr>
<tr><td colspan="3">Time Step Variation = Piecewise Constant</td></tr>
<tr><td><Variation Function></td><td>string</td><td>Piecewise Constant</td></tr>
<tr><td><Variation Value></td><td>list</td><td>S0 value and unit and dataset name (e.g., [\"1s\", \"pwl(ds1,Time)\"])</td></tr>
<tr><td colspan="3">Time Step Variation = Piecewise Linear</td></tr>
<tr><td><Variation Function></td><td>string</td><td>Piecewise Linear</td></tr>
<tr><td><Variation Value></td><td>list</td><td>S0 value and unit and dataset name (e.g., [\"1s\", \"pwl(ds1,Time)\"])</td></tr>
<tr><td colspan="3">Time Step Variation =Automatic</td></tr>
<tr><td><Variation Function></td><td></td><td>Automatic</td></tr>
<tr><td><Variation Value></td><td></td><td>Comma-separated values and units: (No of Fixed Time Steps, Min Time step size, Max Time step size, Min Step Change Factor, Max Step Change Factor, Error Tolerance)</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>InsertSetup (<NAME>, <Enabled>, <Flow Regime>, <Turbulent Model Eqn>, <Include Temperature>, <Include Flow>, <Include Gravity>, <Solution Initialization - X Velocity>, <Solution Initialization - Y Velocity>, <Solution Initialization - Z Velocity>, <Solution Initialization - Temperature>, <Solution Initialization - Turbulent Kinetic Energy>, <Solution Initialization - Turbulent Dissipation Rate>, <Solution Initialization - Specific Dissipation Rate>, <Convergence Criteria - Flow>, <Convergence Criteria - Energy>, <Convergence Criteria - Turbulent Kinetic Energy>, <Convergence Criteria - Turbulent Dissipation Rate>, <Convergence Criteria - Specific Dissipation Rate>, <Convergence Criteria - Discrete Ordinates>, <IsEnabled>, <Radiation Model>, <Flow Iteration Per Radiation Iteration>, <ThetaDivision>, <PhiDivision>, <ThetaPixels>, <PhiPixels>, <Under-relaxation - Pressure>, <Under-relaxation - Momentum>, <Under-relaxation - Temperature>, <Under-relaxation - Turbulent Kinetic Energy>, <Under-relaxation - Turbulent Dissipation Rate>, <Under-relaxation - Specific Dissipation Rate>, <Discretization Scheme - Pressure>, <Discretization Scheme - Momentum>, <Discretization Scheme - Temperature>, <Secondary Gradient>, <Discretization Scheme - Turbulent Kinetic Energy>, <Discretization</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.

	Scheme - Turbulent Dissipation Rate>, <*Discretization Scheme - Specific Dissipation Rate*>, <*Discretization Scheme - Discrete Ordinates*>, <*Linear Solver Type - Pressure*>, <*Linear Solver Type - Momentum*>, <*Linear Solver Type - Temperature*>, <*Linear Solver Type - Turbulent Kinetic Energy*>, <*Linear Solver Type - Turbulent Dissipation Rate*>, <*Linear Solver Type - Specific Dissipation Rate*>, <*Linear Solver Termination Criterion - Pressure*>, <*Linear Solver Termination Criterion - Momentum*>, <*Linear Solver Termination Criterion - Temperature*>, <*Linear Solver Termination Criterion - Turbulent Kinetic Energy*>, <*Linear Solver Termination Criterion - Turbulent Dissipation Rate*>, <*Linear Solver Termination Criterion - Specific Dissipation Rate*>, <*Linear Solver Residual Reduction Tolerance - Pressure*>, <*Linear Solver Residual Reduction Tolerance - Momentum*>, <*Linear Solver Residual Reduction Tolerance - Temperature*>, <*Linear Solver Residual Reduction Tolerance - Turbulent Kinetic Energy*>, <*Linear Solver Residual Reduction Tolerance - Turbulent Dissipation Rate*>, <*Linear Solver Residual Reduction Tolerance - Specific Dissipation Rate*>, <*Linear Solver Stabilization - Pressure*>, <*Linear Solver Stabilization - Temperature*>, <*Frozen Flow Simulation*>, <*Sequential Solve of Flow and Energy Equations*>, <*Convergence Criteria - Max Iterations*>)

Python Example	`oModule.InsertSetup("IcepakTransient",`
	`[`
	`"NAME:Setup1",`
	`"Enabled:=" , True,`
	`"Flow Regime:=" , "Laminar",`
	`"Include Temperature:=" , True,`
	`"Include Flow:=" , True,`
	`"Include Gravity:=" , False,`
	`"Solution Initialization - X Velocity:=", "0m_per_sec",`
	`"Solution Initialization - Y Velocity:=", "0m_per_sec",`
	`"Solution Initialization - Z Velocity:=", "0m_per_sec",`
	`"Solution Initialization - Temperature:=", "AmbientTemp",`
	`"Solution Initialization - Turbulent Kinetic Energy:=", "1m2_per_s2",`

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

```
                "Solution Initialization - Turbulent Dissipation Rate:=", "1m2_per_s3",
                "Solution Initialization - Specific Dissipation Rate:=", "1diss_per_s",
                "Convergence Criteria - Flow:=", "0.001",
                "Convergence Criteria - Energy:=", "1e-07",
                "Convergence Criteria - Turbulent Kinetic Energy:=", "0.001",
                "Convergence Criteria - Turbulent Dissipation Rate:=", "0.001",
                "Convergence Criteria - Specific Dissipation Rate:=", "0.001",
                "Convergence Criteria - Discrete Ordinates:=", "1e-06",
                "Convergence Criteria - Joule Heating:=", "1e-07",
                "IsEnabled:="           , False,
                "Radiation Model:="     , "Off",
                "Solar Radiation Model:=", "Solar Radiation Calculator",
                "Solar Enable Participating Solids:=", False,
                "Solar Radiation - Scattering Fraction:=", "0",
                "Solar Radiation - North X:=", "0",
                "Solar Radiation - North Y:=", "0",
                "Solar Radiation - North Z:=", "1",
                "Solar Radiation - Day:=", 1,
                "Solar Radiation - Month:=", 1,
                "Solar Radiation - Hours:=", 0,
                "Solar Radiation - Minutes:=", 0,
                "Solar Radiation - GMT:=", "0",
                "Solar Radiation - Latitude:=", "0",
                "Solar Radiation - Latitude Direction:=", "East",
                "Solar Radiation - Longitude:=", "0",
                "Solar Radiation - Longitude Direction:=", "North",
                "Solar Radiation - Ground Reflectance:=", "0",
                "Solar Radiation - Sunshine Fraction:=", "0",
                "Under-relaxation - Pressure:=", "0.3",
                "Under-relaxation - Momentum:=", "0.7",
```


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

```
rst",
                "Discretization Scheme - Turbulent Dissipation Rate:=", "First",
                "Discretization Scheme - Specific Dissipation Rate:=", "First",
                "Discretization Scheme - Discrete Ordinates:=", "First",
                "Linear Solver Type - Pressure:=", "V",
                "Linear Solver Type - Momentum:=", "flex",
                "Linear Solver Type - Temperature:=", "F",
                "Linear Solver Type - Turbulent Kinetic Energy:=", "flex",
                "Linear Solver Type - Turbulent Dissipation Rate:=", "flex",
                "Linear Solver Type - Specific Dissipation Rate:=", "flex",
                "Linear Solver Type - Joule Heating:=", "flex",
                "Linear Solver Termination Criterion - Pressure:=", "0.1",
                "Linear Solver Termination Criterion - Momentum:=", "0.1",
                "Linear Solver Termination Criterion - Temperature:=", "0.1",
                "Linear Solver Termination Criterion -
                 Turbulent Kinetic Energy:=", "0.1",
                "Linear Solver Termination Criterion -
                 Turbulent Dissipation Rate:=", "0.1",
                "Linear Solver Termination Criterion -
                 Specific Dissipation Rate:=", "0.1",
                "Linear Solver Termination Criterion - Joule Heating:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Pressure:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Momentum:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Temperature:=", "0.1",
                "Linear Solver Residual Reduction Tolerance -
                 Turbulent Kinetic Energy:=", "0.1",
                "Linear Solver Residual Reduction Tolerance -
                 Turbulent Dissipation Rate:=", "0.1",
                "Linear Solver Residual Reduction Tolerance -
                 Specific Dissipation Rate:=", "0.1",
                "Linear Solver Residual Reduction Tolerance - Joule Heating:=", "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.

PasteSetup

Use: Paste a solve setup.

Syntax: PasteSetup

Return Value: None

RenameSetup

Renames an existing solution setup.

UI Access	Right-click a solution setup in the Project Manager and then click **Rename** on the shortcut menu.
Parameters	Name / Type / Description <*OldSetupName*> / String / Name of the solution setup being renamed. <*NewSetupName*> / String / New name for the solution setup.
Return Value	None.

Python Syntax	RenameSetup (<*OldSetupName*>, <*NewSetupName*>)
Python Example	`oModule.RenameSetup("Setup1", "MySetup")`

RevertAllToInitial

Marks the current mesh for all solution setups as invalid. This will force the next simulation to begin with the initial mesh.

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	> **Analysis Setup** > **Revert to Initial Mesh**.
Parameters	None.
Return Value	None.

Python Syntax	RevertAllToInitial ()
Python Example	`oModule.RevertAllToInitial()`

RevertSetupToInitial

Marks the current mesh for a solution setup as invalid. This will force the next simulation to begin with the initial mesh.

UI Access	Right-click a solution setup in the Project Manager and then click **Rename** on the shortcut menu.
Parameters	Name | Type | Description <*SetupName*> | String | Name of specified setup.
Return Value	None.

Python Syntax	RevertSetupToInitial (<*SetupName*>)
Python Example	`oModule.RevertSetupToInitial("Setup1")`

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

RevertSetupToInitialCondition

Reverts one or more specified setups to their initial condition. Used for linked thermal designs to revert the target solution (clearing restart, thermal monitor, and field results).

UI Access	**Project Manager** > **Analysis** > right-click ***SetupName*** > **Revert to Initial Condition**. or, with a specific setup selected under **Analysis** in the Project Manager: **Icepak** > **Analysis** > **Revert to Initial Condition** (from the menu bar)
Parameters	Name | Type | Description <*SetupName*> | String | Name of solution setup.
Return Value	None

Python Syntax	RevertSetupToInitialCondition(<*setupName*>)
Python Example	`oModule.RevertSetupToInitialCondition('Setup1')`

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.

16 - 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.

EditSetup

EditSetup [Optimization]

EditSetup [Sensitivity]

EditSetup [Statistical]

GetPropNames [Optimetrics]

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

InsertSetup [Parametric]

InsertSetup [Optimization]

InsertSetup [Sensitivity]

InsertSetup [Statistical]

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

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

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.

Python Syntax	CopySetup (<*SetupName*>)
Python Example	`oModule.CopySetup ("OptimizationSetup1")`

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

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><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 <code>true</code>. If it's <code>false</code>, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the <code>AreThereSimulationsRunning</code> 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	DistributedAnalyzeSetup (<*ParametricSetupName*>)
Python Example	`oModule.DistributedAnalyzeSetup("ParametricSetup1")`

EditSetup

Modifies an existing solution setup.

UI Access	Double-click a solution setup in the project tree to modify its settings.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of the solve setup being edited.</td></tr><tr><td><*Attributes*></td><td>Array</td><td>Structured array. `["NAME:<NewSetupName>", <NamedParameters>]` `See the InsertSetup command for details and examples.`</td></tr></table>
Return Value	None.

Python Syntax	EditSetup (<*SetupName*>, <*Attributes*>)
Python Example	`oModule.EditSetup("Setup1",` `[`

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

```
"NAME:NewSetup",
"AdaptiveFreq:=", "1GHz",
"EnableDistribProbTypeOption:=", false,
"SaveFields:=", "true",
"Enabled:=", true,
["NAME:Cap",
        "MaxPass:=", 10,
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30,
        "AutoIncreaseSolutionOrder:=", false,
        "SolutionOrder:=", "Normal"],
["NAME:DC",
        "Residual:=", 1E-005,
        "SolveResOnly:=", false,
        ["NAME:Cond",
                "MaxPass:=", 10,
                "MinPass:=", 1,
                "MinConvPass:=", 1,
                "PerError:=", 1,
```


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

```
                "PerRefine:=", 30),
        ["NAME:Mult",
                "MaxPass:=", 1,
                "MinPass:=", 1,
                "MinConvPass:=", 1,
                "PerError:=", 1,
                "PerRefine:=", 30]],
["NAME:AC",
        "MaxPass:=", 10,
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30]]
)
oModule.EditSetup("HfssDrivenAuto",
["NAME:Setup1",
  "IsEnabled:=", True,
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
```


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

```
            "RangeType:=", "LinearStep",
            "RangeStart:=", "1GHz",
            "RangeEnd:=", "10GHz",
            "RangeStep:=", "1GHz"
            ]
  ],
  "SaveRadFieldsOnly:=", False,
  "SaveAnyFields:=", True,
  "Type:=", "Discrete"
  ])

oModule.EditSetup("AC Magnetic",
[
  "NAME:AC Magnetic",
  "Enabled:="             , True,
  [
    "NAME:MeshLink",
    "ImportMesh:="          , False
  ],
  "MaximumPasses:="       , 4,
  "MinimumPasses:="       , 2,
```


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

```
"MinimumConvergedPasses:=", 1,
"PercentRefinement:="   , 30,
"SolveFieldOnly:="      , False,
"PercentError:="        , 0.1,
"SolveMatrixAtLast:="   , True,
"UseNonLinearIterNum:=" , False,
[
  "NAME:ExpressionCache",
  [
    "NAME:CacheItem",
    "Title:="               , "eddy_loss1",
    "Expression:="          , "eddy_loss",
    "Intrinsics:="          , "Phase=\'0deg\'",
    "ReportType:="          , "Fields",
    [
      "NAME:ExpressionContext"
    ]
  ]
],
"UseCacheFor:="         , ["Pass"],
```


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

```
  "UseIterativeSolver:="  , False,
  "RelativeResidual:="    , 0.0001,
  "NonLinearResidual:="   , 0.0001,
  "SmoothBHCurve:="       , False,
  "Frequency:="           , "200Hz",
  "HasSweepSetup:="       , False,
  "UseHighOrderShapeFunc:=", False,
  "UseMuLink:="           , False,
  "LossAdaptiveCtrl:="    , "0.3"
])
oModule.EditSetup("HfssDriven",
["NAME:Setup3",
        "AdaptMultipleFreqs:=", False,
        "Frequency:=", "5GHz",
        "MaxDeltaS:=", 0.02,
        "PortsOnly:=", False,
        "UseMatrixConv:=", False,
        "MaximumPasses:=", 6,
        "MinimumPasses:=", 1,
        "MinimumConvergedPasses:=", 1,
        "PercentRefinement:=", 30,
```


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

```
        "IsEnabled:=", True,
        "BasisOrder:=", 1,
        "DoLambdaRefine:=", True,
        "DoMaterialLambda:=", True,
        "SetLambdaTarget:=", False,
        "Target:=", 0.3333,
        "UseMaxTetIncrease:=", False,
        "PortAccuracy:=", 2,
        "UseABCOnPort:=", False,
        "SetPortMinMaxTri:=", False,
        "UseDomains:=", True,
        "UseIterativeSolver:=", False,
        "IterativeResidual:=", 1E-06,
        "DDMSolverResidual:=", 0.0001,
        "EnhancedLowFreqAccuracy:=", True,
        "SaveRadFieldsOnly:=", False,
        "SaveAnyFields:=", True,
        "IESolverType:=", "Auto",
        "LambdaTargetForIESolver:=", 0.15,
        "UseDefaultLambdaTgtForIESolver:=", True,
```


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

```
        "SkipIERegionSolveDuringAdaptivePasses:=", True
        "RayDensityPerWavelength:=", 4,
        "MaxNumberOfBounces:="  , 5,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "SkipSBRSolveDuringAdaptivePasses:=", True,
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
])
```

Edit an SBR+ Setup with Fast Frequency Looping

```
oModule.EditSetup("HfssDriven",
        [
                "NAME:Setup1",
                "IsEnabled:="           , True,
                [
                        "NAME:MeshLink",
                        "ImportMesh:="          , False
                ],
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
```


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

```
                "RadiationSetup:="      , "",
                "PTDUTDSimulationSettings:=", "None",
                "FastFrequencyLooping:=", True,
                [
                        "NAME:Sweeps",
                        [
                                "NAME:Sweep",
                                "RangeType:="           , "LinearStep",
                                "RangeStart:="          , "1GHz",
                                "RangeEnd:="            , "10GHz",
                                "RangeStep:="           , "1GHz"
                        ]
                ],
                "ComputeFarFields:="    , True
                "UseSBREnhancedRadiatedPowerCalculation:=", True,
                "IsGOBlockageEnabled:=" , False,
                "GOBlockageSurfaceSelfBlock:=", False
        ])
```

Edit and RF Discharge Setup for HFSS

```
import ScriptEnv
```


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

```
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("coaxbend_discharge_r212")
oDesign = oProject.SetActiveDesign("HFSSDesign60degBendTeflon")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.EditSetup("RFDischarge1",
  [
    "NAME:RFDischarge1",
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
      "Soln:="                , "Setup1 : Sweep",
      [
        "NAME:Params",
        "bend_angle:="          , "bend_angle"
      ],
      "ForceSourceToSolve:="  , True,
```


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

```
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "SourceProduct",
      "ApplyMeshOp:="         , True
    ],
    [
      "NAME:Excitations",
      [
        "NAME:1:1",
        "Magnitude:="           , "1",
        "Phase:="               , "0deg"
      ],
      [
        "NAME:2:1",
        "Magnitude:="           , "0",
        "Phase:="               , "0deg"
      ]
    ],
    [
      "NAME:Frequencies",
      "10GHz"
```


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

```
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Helium"
  ])
```

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><th>Name</th><th>Type</th><th>Description</th></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. **True** - enable setup. **False** - disable setup. </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	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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of existing DesignExplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name</td></tr></table>
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	<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/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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DOEName*></td><td>String</td><td>Name of the Design of Experiments (DOE) setup.</td></tr><tr><td><*FileName*></td><td>String</td><td>Output file name with path.</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	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	Type	Description
	typeName	text string	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	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.		

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 name of the active design, in that order separated by a semicolon.

UI Access	N/A
Parameters	None.
Return Value	String indicating the name 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.

Python Syntax	GetName()
Python Example	design_name = oDesign.GetName()

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.

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

	<*values*>	Array	*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.
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 Analysis Display** dialog box, click the **Result** tab, then select **Table** view, and click on the **Export** button		
Parameters	Name	Type	Description
	<SetupName>	String	Must be one of the existing Optimization, Sensitivity, Statistical, or DesignXplorer setup names
	<FileName>	String	Must be a valid file path and name with extension of csv, tab, dat, or txt..
	[useFullOutputName]	Boolean	Optional: defaulted to false. If set to true values will be printed with units. This parameter is ignored for Optimization and Statistical results.
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	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)`
	`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.

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	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()
Python Example	`oModule = oDesign.GetModule("Optimetrics")` `setupNames = oModule.GetSetupNames()`

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

GetSetupNamesByType [Optimetrics]

Gets a list of Optimetrics setup names by type.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><Optimetrics type></td><td>String</td><td>Examples: parametric, optimization, statistical, sensitivity</td></tr></table>
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	<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></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>; 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
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	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.

UI Access	Set Property value on Optimetrics objects
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.

	Property path	text string	Setup property path. See discussion of Property Path
	new Value	Text String, Number, or Boolean	New value data type is depending on the property type,
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	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		

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	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	<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 be solved</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	SolveSetup (<*SetupName*>)
Python Example	`oModule.SolveSetup ("OptimizationSetup1")`

General Commands Recognized by the Optimetrics Module

Following are general script commands recognized by the **Optimetrics** module:

CopySetup

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

DistrbutedAnalyzeSetup

EditSetup

ExportDXConfigFile

ExportOptimetricsProfile

ExportOptimetricsResult

ExportParametricResults

ExportRespSurfaceMinMaxTable

ExportRespSurfaceRefinePoints

ExportRespSurfaceResponsePoints

ExportRespSurfaceVerificationPoints

ExportDOEResponseCurve

ExportDOEResponseCurveSlices

ExportDOEResponseSurface

ExportDOELocalSensitivity

ExportDOELocalSensitivityCurve

GetOptimetricResult

GetPropNames [Optimetrics]

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

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

ImportSetup

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

SolveSetup [Optimetrics]

SolveAllSetup

CopySetup

Copy the specified Optimetrics setup.

UI Access	NA
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup.
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

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><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><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*>)

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.DistributedAnalyzeSetup("ParametricSetup1")`

EditSetup

Modifies an existing solution setup.

UI Access	Double-click a solution setup in the project tree to modify its settings.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of the solve setup being edited.</td></tr><tr><td><*Attributes*></td><td>Array</td><td>Structured array. `["NAME:<NewSetupName>", <NamedParameters>]` `See the InsertSetup command for details and examples.`</td></tr></table>
Return Value	None.

Python Syntax	EditSetup (<*SetupName*>, <*Attributes*>)

Python Example	`oModule.EditSetup("Setup1",`
	`[`
	`"NAME:NewSetup",`
	`"AdaptiveFreq:=", "1GHz",`
	`"EnableDistribProbTypeOption:=", false,`
	`"SaveFields:=", "true",`
	`"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:Cap",
        "MaxPass:=", 10,
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30,
        "AutoIncreaseSolutionOrder:=", false,
        "SolutionOrder:=", "Normal"],
["NAME:DC",
        "Residual:=", 1E-005,
        "SolveResOnly:=", false,
        ["NAME:Cond",
                "MaxPass:=", 10,
                "MinPass:=", 1,
                "MinConvPass:=", 1,
                "PerError:=", 1,
                "PerRefine:=", 30),
        ["NAME:Mult",
                "MaxPass:=", 1,
                "MinPass:=", 1,
```


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

```
                "MinConvPass:=", 1,
                "PerError:=", 1,
                "PerRefine:=", 30]],
["NAME:AC",
        "MaxPass:=", 10,
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30]]
)
oModule.EditSetup("HfssDrivenAuto",
["NAME:Setup1",
  "IsEnabled:=", True,
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
            "RangeType:=", "LinearStep",
            "RangeStart:=", "1GHz",
            "RangeEnd:=", "10GHz",
            "RangeStep:=", "1GHz"
            ]
```


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

```
  ],
  "SaveRadFieldsOnly:=", False,
  "SaveAnyFields:=", True,
  "Type:=", "Discrete"
  ])

oModule.EditSetup("AC Magnetic",
[
  "NAME:AC Magnetic",
  "Enabled:="             , True,
  [
    "NAME:MeshLink",
    "ImportMesh:="          , False
  ],
  "MaximumPasses:="       , 4,
  "MinimumPasses:="       , 2,
  "MinimumConvergedPasses:=", 1,
  "PercentRefinement:="   , 30,
  "SolveFieldOnly:="      , False,
  "PercentError:="        , 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.

```
"SolveMatrixAtLast:="   , True,
"UseNonLinearIterNum:=" , False,
[
  "NAME:ExpressionCache",
  [
    "NAME:CacheItem",
    "Title:="               , "eddy_loss1",
    "Expression:="          , "eddy_loss",
    "Intrinsics:="          , "Phase=\'0deg\'",
    "ReportType:="          , "Fields",
    [
      "NAME:ExpressionContext"
    ]
  ]
],
"UseCacheFor:="         , ["Pass"],
"UseIterativeSolver:="  , False,
"RelativeResidual:="    , 0.0001,
"NonLinearResidual:="   , 0.0001,
"SmoothBHCurve:="       , False,
"Frequency:="           , "200Hz",
```


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

```
  "HasSweepSetup:="       , False,
  "UseHighOrderShapeFunc:=", False,
  "UseMuLink:="           , False,
  "LossAdaptiveCtrl:="    , "0.3"
])
oModule.EditSetup("HfssDriven",
["NAME:Setup3",
        "AdaptMultipleFreqs:=", False,
        "Frequency:=", "5GHz",
        "MaxDeltaS:=", 0.02,
        "PortsOnly:=", False,
        "UseMatrixConv:=", False,
        "MaximumPasses:=", 6,
        "MinimumPasses:=", 1,
        "MinimumConvergedPasses:=", 1,
        "PercentRefinement:=", 30,
        "IsEnabled:=", True,
        "BasisOrder:=", 1,
        "DoLambdaRefine:=", True,
        "DoMaterialLambda:=", True,
```


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

```
        "SetLambdaTarget:=", False,
        "Target:=", 0.3333,
        "UseMaxTetIncrease:=", False,
        "PortAccuracy:=", 2,
        "UseABCOnPort:=", False,
        "SetPortMinMaxTri:=", False,
        "UseDomains:=", True,
        "UseIterativeSolver:=", False,
        "IterativeResidual:=", 1E-06,
        "DDMSolverResidual:=", 0.0001,
        "EnhancedLowFreqAccuracy:=", True,
        "SaveRadFieldsOnly:=", False,
        "SaveAnyFields:=", True,
        "IESolverType:=", "Auto",
        "LambdaTargetForIESolver:=", 0.15,
        "UseDefaultLambdaTgtForIESolver:=", True,
        "SkipIERegionSolveDuringAdaptivePasses:=", True
        "RayDensityPerWavelength:=", 4,
        "MaxNumberOfBounces:="  , 5,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "SkipSBRSolveDuringAdaptivePasses:=", True,
```


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

```
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
])
```

Edit an SBR+ Setup with Fast Frequency Looping

```
oModule.EditSetup("HfssDriven",
        [
                "NAME:Setup1",
                "IsEnabled:="           , True,
                [
                        "NAME:MeshLink",
                        "ImportMesh:="          , False
                ],
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
                "RadiationSetup:="      , "",
                "PTDUTDSimulationSettings:=", "None",
                "FastFrequencyLooping:=", 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",
                        [
                                "NAME:Sweep",
                                "RangeType:="           , "LinearStep",
                                "RangeStart:="          , "1GHz",
                                "RangeEnd:="            , "10GHz",
                                "RangeStep:="           , "1GHz"
                        ]
                ],
                "ComputeFarFields:="    , True
                "UseSBREnhancedRadiatedPowerCalculation:=", True,
                "IsGOBlockageEnabled:=" , False,
                "GOBlockageSurfaceSelfBlock:=", False
        ])
```

Edit and RF Discharge Setup for HFSS

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("coaxbend_discharge_r212")
oDesign = oProject.SetActiveDesign("HFSSDesign60degBendTeflon")
```


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("AnalysisSetup")
oModule.EditSetup("RFDischarge1",
  [
    "NAME:RFDischarge1",
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
      "Soln:="                , "Setup1 : Sweep",
      [
        "NAME:Params",
        "bend_angle:="          , "bend_angle"
      ],
      "ForceSourceToSolve:="  , True,
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "SourceProduct",
      "ApplyMeshOp:="         , 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:Excitations",
      [
        "NAME:1:1",
        "Magnitude:="           , "1",
        "Phase:="               , "0deg"
      ],
      [
        "NAME:2:1",
        "Magnitude:="           , "0",
        "Phase:="               , "0deg"
      ]
    ],
    [
      "NAME:Frequencies",
      "10GHz"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
```


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

	"Maximum Pressure:=" , "101325pascal", "Postproc Sampling:=" , 500, "Temperature:=" , "0cel", "BuiltInGas:=" , "Helium"])

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><th>Name</th><th>Type</th><th>Description</th></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. • **True** - enable setup. • **False** - disable setup.</td></tr></table>
Return Value	None.

Python Syntax	EnableSetup(<*SetupName*>, <*Enable*>)
Python Example	`oModule.EnableSetup("OptimizationSetup1", True)`

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

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Must be one of existing DesignExplorer setup names</td></tr><tr><td><FileName></td><td>String</td><td>Must be a valid file path and name</td></tr></table>
Return Value	None

Python Syntax	ExportDXConfigFile (<SetupName>, <FileName>)
Python Example	`oModule.ExportDXConfigFile ("DesignXplorerSetup1",` `"c:/exportdir/DXSetup1.xml")`

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></table>

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

	[profileNum]	String	Must be a numeric string. Optional: defaulted to last profile number. It should be a zero indexed profile number.
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	Name	Type	Description
	<SetupName>	String	Must be one of the existing Optimization, Sensitivity, Statistical, or DesignXplorer setup names
	<FileName>	String	Must be a valid file path and name with extension of csv, tab, dat, or txt..
	[useFullOutputName]	Boolean	Optional: defaulted to false. If set to true values will be printed with units. This parameter is ignored for Optimization and Statistical results.
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	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)`

ExportParametricSetupTable

Exports the parametric setup table as a CSV 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	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/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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DOEName*></td><td>String</td><td>Name of the Design of Experiments (DOE) setup.</td></tr><tr><td><*FileName*></td><td>String</td><td>Output file name with path.</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	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")`

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

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

	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.
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	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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DOEName*></td><td>String</td><td>Name of the Design of Experiments (DOE) setup.</td></tr><tr><td><*FileName*></td><td>String</td><td>Output file name with path.</td></tr></table>
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
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** 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.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** 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

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	*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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DOEName*></td><td>String</td><td>Name of the Design of Experiments (DOE) setup.</td></tr><tr><td><*FileName*></td><td>String</td><td>Output file name with path.</td></tr></table>
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
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** 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.ExportDOELocalSensitivityCurve("DesignOfExperimentsSetup1", "C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_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*>

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.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	Name | Type | Description typeName | text string | 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 | 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.

	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()

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 name of the active design, in that order separated by a semicolon.

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 indicating the name of the active design.

Python Syntax	GetName()
Python Example	design_name = oDesign.GetName()

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.

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><td>Name</td><td>Type</td><td>Description</td></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>
</td></tr>
<tr><td>Return Value</td><td>Calculation result. If the setup contains more than one calculation, the output will be an array of values.</td></tr>
</table>

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.`

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 ()

Python Example	`oOptModule.GetPropNames()`
	`oOptModule.GetPropNames(True)`
	`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

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	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.

UI Access	NA		
Parameters	Name	Type	Description
	<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"])`

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><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.

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 **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>
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	<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 solved</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

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	SolveSetup (<*SetupName*>)
Python Example	`oModule.SolveSetup ("OptimizationSetup1")`

Parametric Script Commands

EditSetup [Parametric]

ExportParametricSetupTable

GenerateVariationData [Parametric]

InsertSetup [Parametric]

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>)

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	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	

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')`

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**

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 to use for the variation data generation</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><Parametric Params></td><td>Array</td><td>`["NAME:<SetupName>", "SaveFields:=",`
	`<SaveField>, <StartingPoint>, "Sim. Setups:=",`
	`<SimSetups>,`
	`<SweepDefs>, <SweepOps>,`
	`["NAME:Goals", ["NAME:Goal",`
	`<OptiGoalSpec>), ... ["NAME:Goal",`
	`<OptiGoalSpec>]]`</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the parametric setup.</td></tr><tr><td><SimSetups></td><td>Array of Strings</td><td>An array of Twin Builder solution setup names.</td></tr><tr><td><SweepDefs></td><td>Array</td><td>`["NAME:Sweeps",`
	`["NAME:SweepDefinition", "Variable:=",`
	`<VarName>, "Data:=", <SweepData>,`</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.

			"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		

Python Syntax	InsertSetup ("OptiParametric", <ParametricParams>)
Python Example	oModule.InsertSetup("OptiParametric", ["NAME:ParametricSetup1", "SaveFields:=", true, ["NAME:StartingPoint"], "Sim. Setups:=", ["Setup1"], [

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",
    [
      "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
    ]
  ],
  [
    "NAME:Sweep Operations"
  ],
  [
    "NAME:Goals",
    [
      "NAME:Goal",
      "Solution:=", "Setup1 : LastAdaptive",
      "Calculation:=", "returnloss",
      "Context:=", ""
      [
        "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:=", "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.

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,`
	`  "SaveSolutions:=", 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: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",
      "Range:=", ["Var:=", "Time","Type:=", "a"]
```


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

```
    ],
    "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	<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>

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])

InsertSetup [Sensitivity]

Inserts a new sensitivity setup.

UI Access	Right-click **Optimetrics** in the project tree, and then click **Add>Sensitivity** on the shortcut menu.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SensitivityParams*></td><td>Array</td><td>["NAME:<SetupName>", "SaveFields:=", <SaveField>, ["Name:StartingPoint"], "MaxIterations:=", <MaxIter>, "PriorPSetup:=", <PriorSetup>, "PreSolvePSetup:=", <Preceed>, <SensitivityVars>, <Constraint>, ["NAME:Goals", ["NAME:Goal",</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.

			<OptiGoalSpec>], ..., ["NAME:Goal", <OptiGoalSpec>]], "Primary Goal:=". <PrimaryGoalID>, "PrimaryError:=", <PrimaryError>]
	<*SensitivityVars*>	Array	["NAME:Variables", "VarName:=", Array["i:=", <IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "IDisp:=", <InitialDisp>],... "VarName:=", ["i:=", <IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "IDisp:=", <InitialDisp>]]
	<*InitialDisp*>	VarValue	Index of the Primary goal. Index starts from zero.
	<*PrimaryError*>	Double	Error associated with the Primary goal.
Return Value	None		

Python Syntax	InsertSetup ("OptiSensitivity", <SensitivityParams>,<*SensitivityVars*>, <*InitialDisp*>,<*PrimaryError*>)

Python Example	`oModule.InsertSetup(`
	`"OptiSensitivity",`
	`["NAME:SensitivitySetup1",`
	`"SaveFields:=", true,`
	`["NAME:StartingPoint"],`
	`"MaxIterations:=", 20,`

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

```
"PriorPSetup:=", "",
"PreSolvePSetup:=", true,
["NAME:Variables"],
["NAME:LCS"],
"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:=", ["Var:=","Freq",
```


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

	"Type:=", "s", "Start:=", "8GHz", "Stop:=", "8GHz"]]]], "Primary Goal:=", 1, "PrimaryError:=", 0.001])

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	Name | Type | Description <SetupName> | String | Name of the Setup | |
Return Value	None

Python Syntax	EditSetup (<SetupName>, <StatisticalParams>)
Python Example	`See EditSetup [Optimization]`

InsertSetup [Statistical]

Inserts a new statistical 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 **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:=", <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

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

			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",`
	`  "Calculation:=", "returnloss",`
	`  "Context:=", "",`
	`  ["NAME:Ranges",`
	`  "Range:=", ["Var:=", "Freq",`
	`  "Type:=", "s",`
	`  "Start:=", "8GHz", "Stop:=", "8GHz"]]],`
	`  ["NAME:Goal",`
	`  "Solution:=", "Setup1 : LastAdaptive",`
	`  "Calculation:=", "reflect",`

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

	"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.

17 - Solutions Module Script Commands

Solutions commands should be executed by the "Solutions" module.

```
oModule = oDesign.GetModule("Solutions")
```

```
oModule.CommandName <args>
```

DeleteSolutionVariation

ExportNetworkData

ExportTransientData

GetAdaptiveSettings

GetAllSourceMagnitudes

GetAllSourceModes

GetAllSourcePhases

GetAllSources

GetAntennaParameters

GetFaceMeshCentroidAndArea

GetFaceMeshIDAndArea

GetFieldType

GetIncludePortPostProcessing

GetLayerAndNetOfMeshBody

GetMultipactionBreakdown

GetNetworkDataSolution

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

GetNetworkDataSolutionDefinition

GetNetworkPostprocSetup

GetTerminalExcitationType

GetTransientSolveTimes

TDROnReport

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	Name	Type	Description
	<*SoluParams*>	Array	Structured array. `[<DataSpecifierArray>, ...]`
	<*DataSpecifierArray*>	Array	Structured array. `[<DesignVariationKey>, <SetupName>, <SolnName>]`
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"]])

EditSources

Indicates which source excitations should be used for fields post-processing.

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.
	<*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.

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

			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)`

ExportTransientData

Exports transient solution data to a file.

UI Access	**HFSS** > **Results** > **Solution Data...**, then click **Export**.		
Parameters	Name	Type	Description
	<SoluName>	String	Name of specified solution.
	<Variation>	String	Design variation key. Pass empty string for the current nominal variation.
	<FileName>	String	Full path of output file.
	<DataType>	String	Optional. Type of output data.
	<DataFormat>	String	Optional. Format of output data. .csv - Comma Delimited Data. .tab - Tab Delimited Data files.

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

			.dat - Ansys Plot Data files .txt - Post Processing Files
Return Value	None.		

Python Syntax	ExportTransientData(<*SoluName*>, <*Variation*>, <*FileName*>)
Python Example	`oModule.ExportTransientData("Setup1:Transient", "", "C:/transient_solu1.csv")`

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
	<*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")`

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

GetAdaptiveSettings

Obtains the adaptive frequency settings of a specified setup.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of specified setup.
Return Value	Array of stings represents adaptive frequency settings.

Python Syntax	GetAdaptiveSettings(<*SetupName*>)
Python Example	`oModule.GetAdaptiveSettings("Setup1")`

GetAllSourceMagnitudes

Obtains the magnitude values of all defined sources.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing magnitude 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	GetAllSourceMagnitudes()
Python Example	`oModule.GetAllSourceMagnitudes()`

GetAllSourceModes

Obtains mode information of all defined sources.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing mode information.

Python Syntax	GetAllSourceModes()
Python Example	`oModule.GetAllSourceModes()`

GetAllSourcePhases

Obtains the phase values of all defined sources.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing phase 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	GetAllSourcePhases()
Python Example	`oModule.GetAllSourcePhases()`

GetAllSources

Retrieves all sources defined in the current solution setup.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing source names.

Python Syntax	GetAllSources()
Python Example	`oModule.GetAllSources()`

GetAntennaParameters

Retrieves antenna parameters defined in a specified setup.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of specified 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	Array of strings containing antenna parameters.

Python Syntax	GetAntennaParameters(<*SetupName*>)
Python Example	`oModule.GetAntennaParameters("Setup1")`

GetFaceMeshCentroidAndArea

Returns the mesh centroid and area of the specified face for the specified variation.

UI Access	N/A
Parameters	None.
Return Value	Mesh centroid and area of the specified face for the specified variation.

Python Syntax	GetFaceMeshCentroidAndArea()
Python Example	`oModule.GetFaceMeshCentroidAndArea("["Nominal",34]")`

GetFaceMeshIDAndArea

Returns the mesh ID number and area of the specified face for the specified variation.

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	Mesh ID number and area of the specified variation and face.

Python Syntax	GetFaceMeshIDAndArea()
Python Example	`oModule.GetFaceMeshIDAndArea("["Nominal",34]")`

GetFieldType

Gets the field type of a driven modal design solution.

UI Access	N/A
Parameters	None.
Return Value	String containing field type.

Python Syntax	GetFieldType()
Python Example	`oModule.GetFieldType()`

GetIncludePortPostProcessing

Determines whether the Include Port Post Processing Effects option is enabled.

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. • 1 - Include Port Post Processing Effects is enabled. • 0 - Include Port Post Processing Effects is disabled.

Python Syntax	GetIncludePortPostProcessing()
Python Example	`oModule.GetIncludePortPostProcessing()`

GetMultipactionBreakdown

Gets multipaction breakdown from the specified setup.

UI Access	N/A
Parameters	Name / Type / Description <*SetupName*> / String / Name of specified setup. <*Variation*> / String / Design variation. Pass empty string for the current nominal variation.
Return Value	Array of string containing the multipaction breakdown.

Python Syntax	GetMultipactionBreakdown(<*SetupName*>, <*Variation*>)
Python Example	`oModule.GetMultipactionBreakdown("Setup1", "")`

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

GetNetworkDataSolution

Gets matrix data solution from a specified setup.

UI Access	N/A
Parameters	Name | Type | Description <*SoluName*> | String | Name of solution. <*VariationKey*> | String | Design variation key. Pass empty string for the current nominal variation.
Return Value	String containing matrix data.

Python Syntax	GetNetworkDataSolution(<*SoluName*>, <*VariationKey*>)
Python Example	`oModule.GetNetworkDataSolution("Setup1:Sweep1", "")`

GetNetworkDataSolutionDefinition

Gets definition of network data solution from a specified setup.

UI Access	N/A
Parameters	Name | Type | Description <*SoluName*> | String | Name of specified solution.
Return Value	String containing network data 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	GetNetworkDataSolutionDefinition(<*SoluName*>)
Python Example	`oModule.GetNetworkDataSolutionDefinition("Setup1:Sweep1")`

GetNetworkPostprocSetup

Obtains network postprocessing setup.

UI Access	N/A
Parameters	None.
Return Value	String containing postprocessing setup.

Python Syntax	GetNetworkPostprocSetup()
Python Example	`oModule.GetNetworkPostprocSetup()`

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

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	GetSourceContexts()
Python Example	`oModule.GetSourceContexts()`

GetTerminalExcitationType

Obtains the type for terminal excitation in a driven terminal design.

UI Access	N/A
Parameters	None.
Return Value	String containing excitation type.

Python Syntax	GetTerminalExcitationType()
Python Example	`oModule.GetTerminalExcitationType()`

GetTransientSolveTimes

Gets the transient solution solve time.

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.

	<*SoluName*>	String	Name of specified solution.
	<*Variation*>	String	Design variation. Pass empty string for the current nominal variation.
Return Value	Array of strings containing solve time.		

Python Syntax	GetTransientSolveTimes(<*SoluName*>, <*Variation*>)
Python Example	`oModule.GetTransientSolveTimes("Setup1:Transient", "")`

SetSourceContexts

For Near or Far Field projects for Driven Modal or Driven Terminal Network Analysis Solutions, specifies the port name and all modes/terminals of that port to be enabled as Source Context.

UI Access	**HFSS** > **Fields** > **Edit Sources**.		
Parameters	Name	Type	Description
	<*SourceId*>	Array	Name of modes/terminals to be set as source context.
Return Value	None.		

Python Syntax	SetSourceContexts(<*SourceId*>)

Python Example	`oModule.SetSourceContexts(`
	`[`
	`"Box1_T1", "Box1_T2", "Box1_T3", "Current1", "IncPWave1"`
	`])`

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

TDROnReport

Performs a time-domain reflectometry (TDR) analysis on a specified report.

UI Access	Right-click on **Resutls** > **Perform TDR On Report...**
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of specified report.
<*IsStep*>	Boolean	Specifies the input signal as Step or Impulse.
<*RiseTime*>	Double	Specifies rise time with unit in 'ps'.
<*WindowName*>	String	Name of window to apply. Possible values are "Rectangular", "Bartlett", "Blackman", "Hamming", "Hanning", "Kaiser", "Welch".
<*WindowWidth*>	Integer	Width of window in percentage.
<*KaiserParams*>	Double	Optional. If Kaiser window is specified, set corresponding parameters.

Return Value	None.

Python Syntax	TDROnReport(<*ReportName*>, <*IsStep*>, <*RiseTime*>, <*WindowName*>, <*WindowWidth*>, [Optional <*KaiserParams*>])
Python Example	`oModule.TDROnReport("Rept2DRectFreq", True, 1E-12, "Hanning", 100, 1)`

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

18 - 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>
```

AddMarkerToPlot

DeleteFieldPlot

EditSurfaceMeshSummaryData

ExportPlotImageWithViewToFile

ExportSurfaceMeshSummary

ModifyInceptionParameters

RenamePlotFolder

SetFieldPlotSettings

UpdateAllFieldsPlots

UpdateQuantityFieldsPlots

AddMarker[Fields Reporter]

Adds a marker to the current fields plot.

<table>
<tr><td>UI Access</td><td>Right-click Field Overlays > Plot Fields > Marker > Add Marker.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><Position></td><td>Array</td><td>Array containing X, Y and Z coordinates.</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	AddMarker(<*Position*>)

Python Example	`oModule.AddMarker(`
	`  ["-267.007756778494mil",`
	`  "-640.898759461403mil",`
	`  "9.09494701772928e-13mil"])`

AddMarkerToPlot

Adds a marker to a trace on a named field plot.

UI Access	N/A		
Parameters	Name	Type	Description
	<*Location*>	Array	Array of strings containing X.Y, and Z coordinates for the marker.
	<*PlotName*>	String	Name of the field plot.
Return Value	None		

Python Syntax	AddMarkerToPlot(<*Location*>, <*PlotName*>)

Python Example	`oModule.AddMarkerToPlot(`
	`  ["0.290455877780914in", "-0.616900205612183in",`
	`  "1.77635683940025e-015in"],`

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") oModule.AddMarkerToPlot(["-0.317279517650604in", "1.22481322288513in", "0in"], "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	<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></table>
Return Value	None

Python Syntax	AddNamedExpression(<*ExprName*>, <*FieldType*>)
Python Example	`oModule.AddNamedExpression("Mag_JxE", "Fields")`

CalcOp

Performs a calculator operation.

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	Operation commands like **Mag**, **+**, etc.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OpString*></td><td>String</td><td>The text on the corresponding calculator button.</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OpString*></td><td>String</td><td>The text on the corresponding calculator button.</td></tr></table>
Return Value	None.

Python Syntax	CalcStack(<*OpString*>)
Python Example	`oModule.CalcStack("push")`

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

CalculatorRead

Gets a register file and applies it to the calculator stack.

UI Access	Click **Read...** in the **Fields Calculator** dialog.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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><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></table>

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

	<*SoluNameArray*>	Array	Array of strings containing solution names.
	<*VarArray*>	Array	Array of variable names, value pairs.
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	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><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.
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>

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	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")`

ClearAllMarkers[Fields Reporter]

Clears all markers in the current fields overlay 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 **Field Overlays** > **Plot Fields** > **Marker** > **Clear All**.
Parameters	None.
Return Value	None.

Python Syntax	ClearAllMarkers()
Python Example	`oModule.ClearAllMarkers()`

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

Creates a field plot "Field" or a visual ray trace ("VRT") plot.

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.

UI Access	**Icepak** > **Fields** > **Plot Fields** > <*field_quantity*>
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>	string	Field plot name.
<SolutionName>	string	Name of the solution the field plot's data displays. A space is required on either side of the colon (:) character. If omitted, the plot will not be created.
<UserSpecifyName>	string	Name of the field plot.
<UserSpecifyFolder>	int	Name of the folder the field plot is in in the Project Manager.
<QuantityName>	string	"Pressure", "SurfPressure", "Temperature", "SurfTemperature", "VelocityVectors", "Ux", "Uy", "Uz", "SurfVelocityVectors", "SurfUx", "SurfUy", "SurfUz", "Speed", "ViscosityRatio", "WallYPlus", "TKE", "Epsilon", "Kx", "Ky", "Kz", "KxCentroid", "KyCentroid", "KzCentroid", "VolumeHeatLoss", "SurfaceHeatLoss", "CurrentDensity", "JouleHeatingDensity", "ElectricPotential", "CurrentDensityX", "CurrentDensityY", "CurrentDensityZ", "SurfCurrentDensity", SurfJouleHeatingDensity", "SurfElectricPotential", "SurfCurrentDensityX", "SurfCurrentDensityY", "SurfCurrentDensityZ", "HeatFlowRate", "HeatTransCoeff", "HeatFlux", "MassFlux", "MassFlowRate", "RadiationFlux", "RadiationFlow", "VolumeFlowRate", or "FluidDensity"
<PlotFolder>	string	Name of the folder the field plot is in in the Project Manager.
<StreamlinePlot>	bool	True or False.
<AdjacentSidePlot>	bool	True or False.
<FullModelPlot>	bool	True or False.
<IntrinsicVar>	int	Intrinsic variable frequency and phase.
<PlotGeomInfo>	list	List of geometry included in the field plot.
<FilterBoxes>	list	List of geometry included in the filter box.
<NAME>	string	PlotOnSurfaceSettings
<ShadingType>	int	Please type in description manually
<Filled>	bool	True or False.
<IsoValType>	string	Please type in description manually
<AddGrid>	bool	True or False.
<MapTransparency>	bool	True or False.
<Refinement>	int	0

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

	<Transparency>	int	0
	<SmoothingLevel>	int	0
	<NAME>	string	Arrow3DSpacingSettings
	<ArrowUniform>	bool	True or False.
	<ArrowSpacing>	int	0
	<MinArrowSpacing>	int	0
	<MaxArrowSpacing>	int	0
	<GridColor>	list	Color of the field plot grid.
	<EnableGaussianSmoothing>	bool	True or False.
	<SurfaceOnly>	bool	True or False.
Return Value	None		

Python Syntax	CreateFieldPlot (*<NAME>*, *<SolutionName>*, *<UserSpecifyName>*, *<UserSpecifyFolder>*, *<QuantityName>*, *<PlotFolder>*, *<StreamlinePlot>*, *<AdjacentSidePlot>*, *<FullModelPlot>*, *<IntrinsicVar>*, *<PlotGeomInfo>*, *<FilterBoxes>*, *<NAME>*, *<ShadingType>*, *<Filled>*, *<IsoValType>*, *<AddGrid>*, *<MapTransparency>*, *<Refinement>*, *<Transparency>*, *<SmoothingLevel>*, *<NAME>*, *<ArrowUniform>*, *<ArrowSpacing>*, *<MinArrowSpacing>*, *<MaxArrowSpacing>*, *<GridColor>*, *<EnableGaussianSmoothing>*, *<SurfaceOnly>*)

Python Example	`oModule.CreateFieldPlot(`
	`  [`
	`  "NAME:Temperature2",`
	`  "SolutionName:=" , "Setup1 : SteadyState",`
	`  "UserSpecifyName:=" , 0,`
	`  "UserSpecifyFolder:=" , 0,`
	`  "QuantityName:=" , "Temperature",`
	`  "PlotFolder:=" , "Temperature",`

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

```
                "StreamlinePlot:="      , False,
                "AdjacentSidePlot:="    , False,
                "FullModelPlot:="       , False,
                "IntrinsicVar:="        , "",
                "PlotGeomInfo:="        , [1,"Surface","FacesList",1,"HEAT_SINK"],
                "FilterBoxes:="         , [0],
                [
                        "NAME:PlotOnSurfaceSettings",
                        "ShadingType:="         , 0,
                        "Filled:="              , False,
                        "IsoValType:="          , "Tone",
                        "AddGrid:="             , False,
                        "MapTransparency:="     , True,
                        "Refinement:="          , 0,
                        "Transparency:="        , 0,
                        "SmoothingLevel:="      , 0,
                        [
                                "NAME:Arrow3DSpacingSettings",
                                "ArrowUniform:="        , True,
                                "ArrowSpacing:="        , 0,
                                "MinArrowSpacing:="     , 0,
                                "MaxArrowSpacing:="     , 0
                        ],
                        "GridColor:="           , [255,255,255]
                ],
                "EnableGaussianSmoothing:=", False,
                "SurfaceOnly:="         , True
        ], "Field")
```

DeleteFieldPlot

Deletes one or more field plots.

UI Access	Right-click on one filed plot, select **Delete**.

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><*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*>)
Python Example	`oModule.DeleteMarker(["m1", "m2"])`

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")`

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.

Python Syntax	DeleteUneditablePlot(<*NameArray*>)

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.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.

UI Access	**HFSS** > **Fields** > **Create Surface Mesh Summary**, then click the **Setup** 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

Name	Type	Description
<*Name*>	String	"NAME:SurfaceMeshSummary"
<*SolutionName*>	String	Solution name that has the mesh to be evaluated
<*Variation Name*>	String	Name of design variation
<*MeshRowItems*>	Array	Row data that includes <*ItemPerRow*>, <*EntityPerRow*>, and <*MeshDataPer Item*>
<*ItemPerRow*>	String	"NAME:ItemPerRow"
<*EntityPerRow*>	String	Entity ID
<*MeshDataPer Item*>	Array	Mesh data category and statistics

Return Value

None

Python Syntax

EditSurfaceMeshSummaryData ([[<*Name*>,<*SolutionName*>,<*Variation Name*>,<*MeshRowItems*>,[<*ItemPer-Row*>,<*EntityPerRow*>,<*MeshDataPer Item*>]...]])

Python Example

```
oModule = oDesign.GetModule("FieldsReporter")
oModule.EditSurfaceMeshSummaryData(
  [
  [
    "NAME:SurfaceMeshSummary",
    "SolutionName:="        , "Setup1 : LastAdaptive",
    "Variation:="           , "Nominal",
    [
    "NAME:MeshRowItems",
    [
      "NAME:ItemPerRow",
        "EntityPerRow:="        , "annular_rng",
    [
      "NAME:MeshDataPerItem",
      "Min Edge Len:="        , 0.000166961133900917,
      "Max Edge Len:="        , 0.00145730991671888,
      "Mean Edge Len:="       , 0.000524160923260581,
```


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

```
        "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`
	`([`
	`  "1 + 2 j",`

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"])

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*EdgeName*></td><td>String</td><td>Name of an edge defined in the 3D Modeler editor.</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*LineName*></td><td>String</td><td>Name of a line defined in the 3D Modeler editor.</td></tr></table>
Return Value	None.

Python Syntax	EnterLine(<*LineName*>)
Python Example	`oModule.EnterLine("Line1")`

EnterOutputVar

Enters Output Vars, only valid for Eigenmode problems.

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 **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.

Python Syntax	EnterPoint(<*PointName*>)
Python Example	`oModule.EnterPoint("Point1")`

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

EnterQty

Enters a field quantity.

UI Access	Click **Quantity**, and then select from the list.
Parameters	Name: *<FieldQty>* — Type: String — Description: 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: *<Scalar>* — Type: Double — Description: The real number to enter 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	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**.
Parameters	Name | Type | Description <*SurfName*> | String | Name of a surface 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.

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.

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**.

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><*VarNames*></td><td>Array</td><td>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.</td></tr></table>
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	<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")`

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

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.

UI Access	[product] > **Fields** > **Plot Fields** > **Marker** > **Export Marker Table**.		
Parameters	Name	Type	Description
	<*FileName*>	String	Name of export 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	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	

Name	Type	Description
<*OutputFile*>	String	Name of the output file.
<*Min*>	Array	Minimum values for the coordinate components of the grid system
<*Max*>	Array	Maximum values for the coordinate components of the grid system
<*Spacing*>	Array	Spacing values for the coordinate components of the grid system
<*SolnName*>	String	Name of the simulation setup
<*SolnParameters*>	Array	Array of strings containing setup definitions.
<*ExportOption*>	Array	["NAME:ExportOption" <*IncludePointsInOutput*>: Boolean; specifies whether include points in the output file. <*RefCSName*>: String, specifies the reference coordinate system used by the model <*PtInSI*>: Boolean; if True, points are exported in SI. If False, points are exported in model units

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

			<*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",`
	`  "PtInSI:=" , False,`
	`  "FieldInRefCS:=" , True`
	`  ],`
	`  "Cartesian", ["0mm", "0mm", "0mm"],`

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

	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")`

ExportSurfaceMeshSummary

Use: Exports a Surface Mesh Summary.

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

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.

UI Access	Right-click on report name in the project tree and select **Export Data**.
Parameters	See table below

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
<UnitSpec>	String	For example, “kV, Mhz, yd”
<UseTraceNumberFormat>	Boolean	"True", "False"

Return Value	None.

Python Syntax	ExportToFile(*<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.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
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	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.

HidePolarPlot

Hides a currently visible polar plot.

UI Access	N/A
Parameters	Name: *<PlotName>*; Type: String; Description: Name of the polar plot.
Return Value	None.

Python Syntax	HidePolarPlot(<*PlotName*>)
Python Example	`oModule.HidePolarPlot("rE Plot 1")`

HideRadiatedPlotOverlay

Hides overlay radiation fields plot.

UI Access	N/A
Parameters	Name: *<PlotName>*; Type: String; Description: 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	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	<table><tr><th>Name</th><th>Type</th><th>Description</th></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"])`

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

ModifyFieldPlot

Modifies a plot definition.

UI Access	[product] > **Fields** > **Modify Plot**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*OriginalName*></td><td>String</td><td>Name of the plot to be modified.</td></tr><tr><td><*PlotParams*></td><td>Array</td><td>Array containing modified settings.</td></tr></table>
Return Value	None.

Python Syntax	ModifyFieldPlot(<*OriginalName*>, <*PlotParams*>)

Python Example	<pre>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",</pre>

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

```
        "Filled:=", False,
        "IsoValType:=", "Fringe",
        "SmoothShade:=", True,
        "AddGrid:=", False,
        "MapTransparency:=", True,
        "Transparency:=", 0, _
        "ArrowUniform:=", True,
        "ArrowSpacing:=", 0.100000001490116,
        "GridColor:=", [255, 255, 255]]
])
```

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.

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	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, "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,

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

	"MaxArrowSpacing:=", 0], "GridColor:=", [255,255,255]], "EnableGaussianSmoothing:=", False])

RenameFieldPlot

Renames a plot.

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")

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

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.

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.

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	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.
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)`

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

SetFieldPlotSettings

Sets plot attributes.

<table>
<tr><td>UI Access</td><td>[product] > Fields > Modify Plot Attributes</td></tr>
<tr><td>Parameters</td><td>
<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>,
<PlotOnLineSettings>,
<PlotOnSurfaceSettings>,
<PlotOnVolumeSettings>]
See description of CreateFieldPlot command for details.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	SetFieldPlotSettings(<*PlotName*> <*PlotItemAttributes*>)
Python Example	oModule.SetFieldPlotSettings("Mag_E2", ["NAME:FieldsPlotItemSettings", ["NAME:PlotOnLineSettings", ["NAME:LineSettingsID", "Width:=", 4,

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

```
        "Style:=", "Cylinder"],
    "IsoValType:=", "Tone",
    "ArrowUniform:=", True,
    "NumofArrow:=", 100
    ],
    ["NAME:PlotOnSurfaceSettings",
        "Filled:=", False,
        "IsoValType:=", "Tone",
        "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.

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>[product] > Fields > Modify Plot Attributes</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><PlotFolderName></td><td>String</td><td>Name of the folder with the attributes to modify.</td></tr>
<tr><td><PlotFolderAttributes></td><td>Array</td><td>Structured array.
["NAME:FieldsPlotSettings",
"Real time mode:=", <bool>,
<ColorMapSettings>,
<Scale3DSettings>,
<Marker3DSettings>,
<Arrow3DSettings>]</td></tr>
<tr><td><ColorMapSettings></td><td>Array</td><td>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.>]</td></tr>
<tr><td><Scale3DSettings></td><td>Array</td><td>Structured array.
["NAME:Scale3DSettings",
"m_nLevels:=", <int>,</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.

		"m_autoScale:=", <bool>, "minvalue:=", <double>, "maxvalue:=", <double>, "log:=", <bool>, "IntrinsicMin:=", <double>, "IntrinsicMax:=", <double>]
<Marker3DSettings>	Array	Structured array. ["NAME:Marker3DSettings", "MarkerType:=", <integer>, #9:Sphere #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>,

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

	"ArrowSize:=", <double>]
Return Value	None.

Python Syntax	SetPlotFolderSettings(<*PlotFolderName*>, <*PlotFolderAttributes*>)
Python Example	oModule.SetPlotFolderSettings("E Field1", ["NAME:FieldsPlotSettings", "Real time mode:=", True, ["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",

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

	"ArrowType:=", 1, "ArrowMapSize:=", True, "ArrowMapColor:=", True, "ShowArrowTail:=", True, "ArrowSize:=", 0.25]])

UpdateAllFieldsPlots

Updates the All Fields Plots.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	UpdateAllFieldsPlots()
Python Example	`oModule.UpdateAllFieldsPlots()`

UpdateQuantityFieldsPlots

Updates the Quantity Fields Plots.

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><*FolderName*></td><td>String</td><td>Folder containing the plotted quantities.</td></tr></table>
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.

19 - Fields Calculator Script Commands

Fields Calculator commands should be executed by the Field Overlays module, which is called FieldsReporter in Icepak 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.

UI Access	Click **Add** in the **Fields Calculator** window.
Parameters	<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></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	AddNamedExpression(<*ExprName*>, <*FieldType*>)
Python Example	`oModule.AddNamedExpression("Mag_JxE", "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.

UI Access	Click **Read...** in the **Fields Calculator** dialog.
Parameters	Name | Type | Description <*FileName*> | String | Name of the file to be read.

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

	<*SoluName*>	String	Specified solution to read in.
	<*VarArray*>	Array	Array of variables to read.
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	Name	Type	Description
	<*FileName*>	String	Path to and including name of input register file.
	<*SoluName*>	String	Specified solution to read in.
	<*FieldType*>	String	Type of specified field.
	<*VarArray*>	Array	Array of variable names, value pairs.
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	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.

UI Access	Click **Write...** 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><*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><td>Name</td><td>Type</td><td>Description</td></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	Name / Type / Description <*PointName*> / String / Name of a point defined in the 3D Modeler editor.
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	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")`

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: <*VarNames*> — Type: Array — Description: 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.

UI Access	In the Project Manager tree, right click **Field Overlays**, and select **Calculator**
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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 "Fields".</td></tr></table>
Return Value	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".

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)', 'coEnergy = Smooth(CoEnergy)', 'appEnergy = Smooth(App_Energy)', 'Ohmic_Loss = Smooth (Ohmic-Loss)', 'Total_Loss = Smooth(TotalLoss)', 'Temperature = Smooth(Temp)', 'Volume_Force_Density = <VolumeForceDensityx,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.

20 - Fields Summary Script Commands

Fields Summary commands should be executed by the Field Overlays module, which is called "FieldsReporter" in scripts.

```
oModule = oDesign.GetModule("FieldsReporter")

oModule.CommandName <args>
```

EditFieldsSummarySetting

ExportFieldsSummary

EditFieldsSummarySetting

Creates a fields summary report in an Icepak design.

UI Access	**Icepak** > **Fields** > **Create Fields Summary**		
Parameters	Name	Type	Description
	<SolutionName>	string	Name of solution
	<Variation>	string	Name of variation or All
	<Calculation>	list	"*Entity Type*", "*Geometry Type*", "*Selected Geometry*", "*Selected Quantity*", " *Normal*", "*Side*","*True or False*"
	<Entity Type>	bool	Boundary, Object, or Monitor
	<Geometry Type>	bool	Surface or Volume
	<Selected Geometry>	string	Name of selected geometry
	<Selected Quantity>	string	Name of selected quantity
	<Normal>	int	Coordinate values for direction relative to normal
	<Side>	bool	Default, Adjacent, or Combined
	<True or False>	bool	Include all surface meshes
Return Value	None		

Python Syntax	EditFieldsSummarySetting (*<Calculation>*, *<Calculation>*, *<Calculation>*, *<Calculation>*, *<Calculation>*)

Python Example	`oModule.EditFieldsSummarySetting(`
	`  [`
	`  "SolutionName:="  , "Setup1 : SteadyState",`
	`  "Variation:="  , "Nominal",`

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

```
                "Calculation:="         , ["Bound-
ary","Sur-
face","Grille1","Cur-
rentDens-
ityZ","1.00,0.00,0.00","Default","Reduced","Nominal",False],
                "Calculation:="         ,
["Bound-
ary","Volume","CPU","Tem-
perature","","Default","Reduced","Nominal",False],
                "Calculation:="         ,
["Bound-
ary","Volume","Memory","Tem-
perature","","Default","Reduced","Nominal",False],
                "Calculation:="         , ["Monitor","Volume","Exhaust_
Mon-
itor","Speed","1.00,0.00,0.00","De-
fault","Reduced","Nominal",False]
        ])
```

ExportFieldsSummary

Exports a fields summary report as a CSV file.

UI Access	**Icepak** > **Fields** > **Create Fields Summary** > [**Apply and Export | Export**]
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SolutionName*></td><td>string</td><td>Setup name : SteadyState or Setup name : Transient</td></tr><tr><td><*DesignVariationKey*></td><td>string</td><td>Nominal</td></tr><tr><td><*ExportFileName*></td><td>string</td><td>File path to and name of CSV file</td></tr><tr><td><*IntrinsicValue*></td><td>string</td><td>Intrinsic variable</td></tr></table>
Return Value	None

Python Syntax	ExportFieldsSummary (<*SolutionName*>, <*DesignVariationKey*>, <*ExportFileName*>, <*IntrinsicValue*>)

Python Example	`oModule.ExportFieldsSummary(`
	`  [`
	`  "SolutionName:="  , "Setup1 : SteadyState",`
	`  "DesignVariationKey:="  , "Nominal",`
	`  "ExportFileName:="  , "C:\\Users\\spsander\\OneDrive - A`

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

```
Inc\\Desktop\\AEDT\\R22.1\\summaryReport.csv",
                "IntrinsicValue:="      , ""
        ])
```


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 - 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.

<table>
<tr><td>UI Access</td><td>Right click on the created document in the Project Manager under Results > Documents and click Modify document.</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 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>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

<table>
<tr><td>Python Syntax</td><td>EditDocument(<<i>name</i>>,<<i>data</i>>,<<i>traces</i>>)</td></tr>
<tr><td>Python Example</td><td>

```
oModule.EditDocument("Design Summary",
[
  "NAME:Design Summary",
  "",
  "SysLib",
  "DesignSummary",
  ["NAME:Inputs"]
],
["NAME:DocTraces"]
)
```

</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.

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><separator></td><td>String</td><td>Separator used to convey the directory "level"</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*name*></td><td>String</td><td>Name of the document to be saved.</td></tr><tr><td><*saveTo*></td><td>String</td><td>File path to save the document to.</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*name*></td><td>String</td><td>Name of the document to be saved.</td></tr><tr><td><*saveTo*></td><td>String</td><td>File path to save the document at.</td></tr></table>
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.

22 - 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

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: ["multiply_factor:=", "2.0", "component_name:=", "resistor1"]
	<ProbeSelections>	Array	Name of the probe being specified. Note: this must match a probe name spe-

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

			cified 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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SoluName*></td><td>String</td><td>Name of user defined solution to be edited.</td></tr><tr><td><*NewSoluName*></td><td>String</td><td>New name for the specified user defined solution.</td></tr><tr><td><*LibType*></td><td>String</td><td>Indicates the library where the UDS plugin file is located. This parameter must be one of the following values: "SysLib", "UserLib", "PersonalLib".</td></tr><tr><td><*RelativePath*></td><td>String</td><td>The path of the UDS plugin file relative to the "UserDefinedOutputs" sub-directory of the library specified by <*LibType*>.</td></tr><tr><td><*PropList*></td><td>Array</td><td>Strings specify name-value pairs corresponding to the UDS properties specified in the plugin file. For example: `["multiply_factor:=", "2.0", "component_name:=", "resistor1"]`</td></tr><tr><td><*ProbeSelections*></td><td>Array</td><td>Name of the probe being specified. Note: this must match a probe name specified in the UDS plugin file.</td></tr><tr><td><DynamicProbes></td><td>Array</td><td>Array of <ProbeSelection>'s, representing the probes that are used by dynamic-probes.</td></tr></table>
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.

23 - 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

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>, ...]`

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

		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" 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	

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

ExportSpreadsheet

Writes the S-parameters to disk in spreadsheet format (*.tab text file).

UI Access	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.

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	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**.
<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

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

			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)`

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).

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 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)
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.
	<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 Syntax	ExportTouchstone()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `success = oData.ExportTouchstone("D:\folder\ne133.s2p", 0, 1, 12, [], True, 23, "GHz")`

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.

ExportTouchstone2

Writes the S-parameters to disk in Touchstone 2 format (*.ts 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 2 Format (*.ts) 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>
</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.

	<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 Syntax	ExportTouchstone2()

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.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*).
Return Value	Network IDispatch.

Python Syntax	Extract()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Extract(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.

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()`

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.

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	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.

UI Access	None.
Parameters	None.
Return Value	String name.

Python Syntax	GetName()
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.

	`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")` `num = oData.GetPortCount()`

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.

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")` `index = oPostProc.GetPortNumber("Input35")`

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.

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()`

> **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.

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** *<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.

UI Access	None.
Parameters	None.
Return Value	String variation key.

Python Syntax	GetVariation()
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.

	`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 • 3 – Gamma • 4 – Impedance **Note:** The default value is **0**.
<comparePortNames>	Boolean	If **True**, port names must be identical.

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 **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()
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.

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

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")`

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.

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 **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.

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.

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

	<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.

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.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Renormalize** 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.

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

	<portImpedances>	Array of Doubles	The new impedance values to be applied to each port.
Return Value	Network IDispatch. **Note:** Renormalize returns **None** if there is not an impedance value for each port.		

Python Syntax	Renormalize()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Renormalize(oData1, [3, 2])`

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.

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	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.

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.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Reorder** 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 in new order (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. **Note:** Reorder returns **None** if the array argument does not contain all the terminal port numbers, in any order.

Python Syntax	Reorder()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Reorder([3, 2, 1])`

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.

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	Reset()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oPostProc.Reset()`

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.

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	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.

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")`

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.

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** window. 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")`

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.

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.

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.

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

Smooth

Creates a new network data with values smoothed. The number of adjacent points smoothed is user-specified.

<table>
<tr><td>UI Access</td><td>From the Network Data Explorer tab, select Transforms in the NDE ribbon. Then select Smooth 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><order></td><td>Integer</td><td>The number of adjacent points that are smoothed.
Note: The default value is 10.</td></tr>
<tr><td><enforceCausality></td><td>Boolean</td><td>Casual data is calculated before smoothing.
Note: The default value is False.</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>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, <code>enforceCausality</code>.</td></tr>
</table>

Python Syntax	Smooth()

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")` `oData2 = oNDE.Smooth(oData1, 3, True)`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

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.

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	Stretch()
Python Example	`oNDE = oDesktop.GetTool("ndExplorer")` `oData2 = oNDE.Stretch(oData1, 3.1)`

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.

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

	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")` `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.

24 - 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

AreMaterialPropertiesEqual

AreSurfaceMaterialPropertiesEqual

CloneMaterial

DeleteDataset

DoesMaterialExist

EditDataset

ExportDataset

ExportMaterial

GetProjectMaterialNames

GetPropertyValue

ImportDataset

RemoveMaterial

RemoveUnusedDefinitions

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

SetPropertyValue

UpdateDefFromBlock

UpdateDefFromBlockEx

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.

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

	<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.
	<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(`

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

```
["NAME:Component",
  "Info:=", ["Type:=", 0,
    "NumTerminals:=", 0,
    "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,
```


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

	"GroupID:=" , 0], "CircuitEnv:=", 0, "Refbase:=" , "U", "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*>

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.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.

	])

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*defBlock*></td><td>String</td><td>Text of the new material definition in block form.</td></tr><tr><td><*defFolderName*></td><td>String</td><td>Library type (by definition folder name)</td></tr><tr><td><*newTimeStamp*></td><td>String</td><td>New timestamp (time_t as integer number of seconds since 1/1/1970 12:00am, as string), default is current time</td></tr><tr><td><*replaceExisting*></td><td>Boolean</td><td>True to replace existing, False to choose a new unique name if an existing definition is found</td></tr></table>
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",`

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

```
    "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()
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'"
```


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.AddDefinitionFromBlock(defBlock, "Materials",
"10101010", True)
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)
materialNameInQuotes = "\"" + addedName + "\""
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers",
        "Box1"
      ],
      ["NAME:ChangedProps",
        ["NAME:Material",
         "Value:=", materialNameInQuotes
        ]
      ]
    ]
  ])
```

AddMaterial

Adds a local material.

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>Add Material in the material editor.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><MaterialParams></td><td>Array</td><td>[
"NAME: <name of the material to be added>",
<MatProperty>, <MatProperty>, ...
]</td></tr>
<tr><td><MatProperty></td><td>Array</td><td>For simple material:
"<PropertyName>:=", <value>

For anisotropic material:
[
"NAME:<PropertyName>",
"property_type:=", "AnisoProperty",
"unit:=", <Unit>",
"component1:=", <value>,
"component2:=", <value>,
"component3:=", <value>))
]</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.

	<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): 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",`

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

```
    "component1:=", "11",
    "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
```


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
"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",`

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",
      "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:=", "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

<TextInfo>)

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

<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>,

<bool>, // is plotter font

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

<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

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')`

Edit [component manager]

Modifies an existing 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><th>UI Access</th><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 any or all of the following:

"VariableProp:=", <VariableInfo>,
"CheckboxProp:=", <CheckBoxInfo>,
"ButtonProp:=", <ButtonInfo>,</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.

			"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.		

<TypeInfo>:

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

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

Simplorer:
# choices include "Electrical", "Magnetic", "Fluidic", "Translational",
"Translational_V", "Rotational", "Rotational_V",
```


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

```
""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
<FlagLetters>,
<string>, # description
<string>, # button title
<string>, # extra text
```


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

```
<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
```

<NumberInfo>:

```
[<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
<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>,
<string>, # description
"CB:=", <string>, # optional - script for call back
<string>, # x value: number with length units
```


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

```
<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
```

<CosimDefInfo>:

```
["NAME:CosimDefinition",
"CosimulatorType:=", <int>,
```


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

```
"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",`
	`"ModifiedOn:=", 1152722112,`
	`"Manufacturer:=", "",`
	`"Symbol:=", "nexx_bjt_npn",`

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

```
      "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"],
  ["NAME:Parameters",
    "TextProp:=", ["LabelID","HD", "Property string for netlist ID", "Q@ID"],
    "TextProp:=", ["MOD", "D","Name of model data reference", "required"],
```


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

```
    "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:=", "",

    "ModifiedOn:=", 1071096503,

    "Manufacturer:=", "Ansys",

    "Symbol:=", "bendo",
```


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

```
"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",
  "MenuProp:=", ["CoSimulator", "D","", "Default, Custom, Netlist", 0],
  "ButtonProp:=", ["CosimDefinition",  "D", "", "", "Edit", 0, "But-
  tonPropClientData:=", []]
```


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

```
    ],
    ["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",
        "CosimulatorType:=", 3,
        "CosimDefName:=", "Netlist",
```


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, "NetlistString:=", ""]]])

EditDataset

Modifies a dataset. This can be executed by the oProject, or oDesign variables.

UI Access	**Project** > **Datasets** > **Edit**.
Parameters	<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>
Return Value	None.

Python Syntax	EditDataset (<*OriginalName*> <*DatasetDataArray*>)
Python Example	oProject.EditDataset ("ds1" ["NAME:ds2", ["NAME:Coordinates", ["NAME:Coordinate", "X:=", 1, "Y:=", 2],

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

```
    [
      "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.

<table>
<tr><td>UI Access</td><td>View/Edit Materials command in the material editor</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 material before editing.</td></tr>
<tr><td><MatProperties></td><td>Array</td><td>Structured array containing material properties:
<pre>
[
  "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>,
</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.

		"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>, "prop_modifier:=" , <"thermal_modifier" or "spatial_modifier">,

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

	"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", "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.

```
  "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",
"conductivity:=" , "58000000",
"thermal_conductivity:=" , "400",
```


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:=" , "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",
      "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: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
      ]
    ]
  ])
```

Edit [padstack manager]

Use: Edit an existing padstack.

Command: Tools > Edit Configured Libraries > Padstacks > Edit Padstack

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

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

"sb2:=", <string>, // solderball mid diameter, real with units

"sbn:=", <string>), // name of solderball material

"ppl:=", <PadPortLayerArray>)

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> // 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",

"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>:

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

"Rct" // Rectangle

"Ov" // Oval

"Blt" // Bullet

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

"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

"135" // 135 degrees

"180" // 180 degrees

"225" // 225 degrees

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

"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

"blw" // below padstack

<PadPortLayerArray>:

Array(<int>, <int>,....) where each int is a 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.

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.

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 foot-

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

			print 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 allowed.
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	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	<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 pad-stack to <*PadstackName*> – String that defines the simple name of the padstack 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(*[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.

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**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	Includes the following: <*LibraryName*> – String that defines the name of the library to export the symbol to

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

			<*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")`

GetProjectMaterialNames

Returns the material names belonging to an active project.

UI Access	N/A		
Parameters	Name	Type	Description
	None		
Return Value	String names of the materials in the active project.		

Python Syntax	GetProjectMaterialNames()
Python Example	`oProject = oDesktop.GetActiveProject()`

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() 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

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")

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

			• 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 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**.

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
	<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>*)

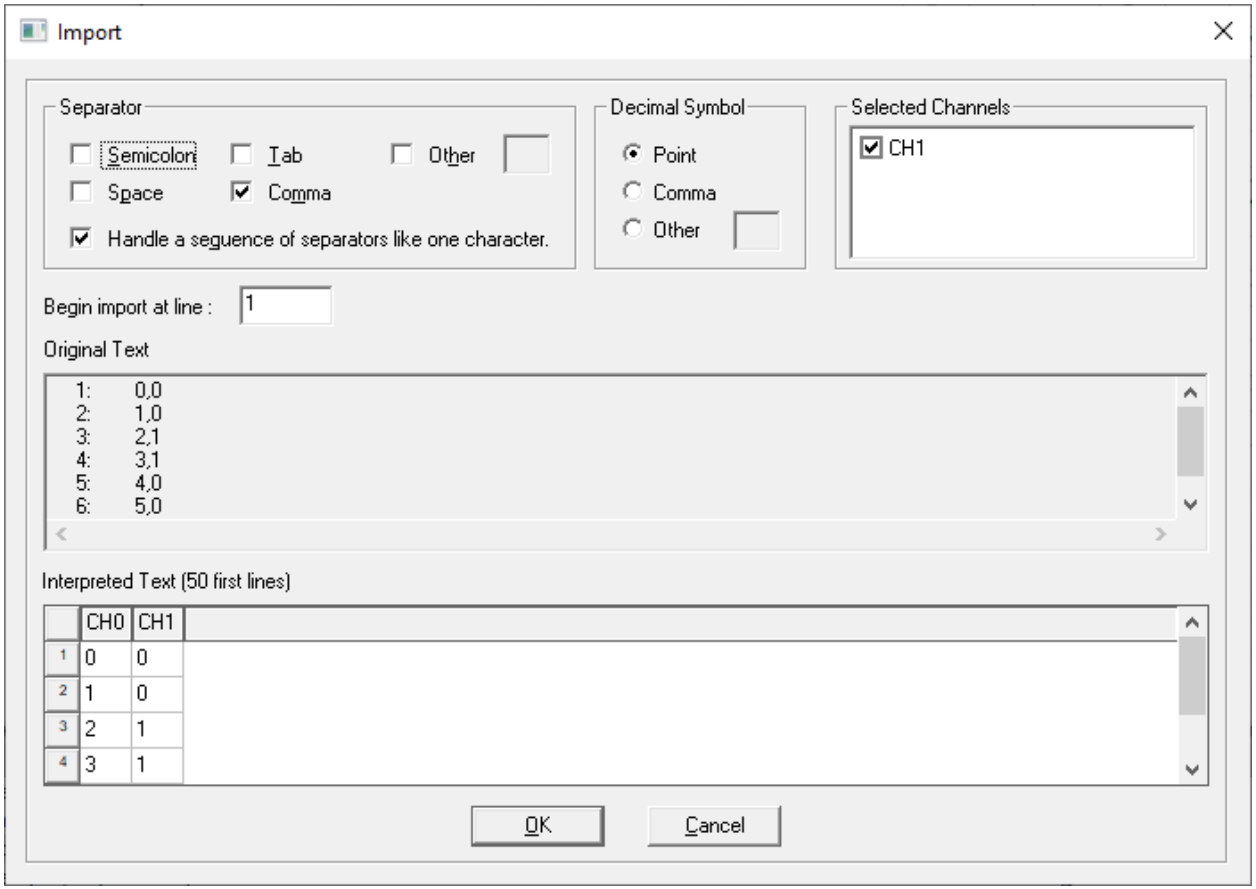
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab')`
	`oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`
	`oDesign.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></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"])`

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")
```

Material Manager Script Commands

The material manager provides access to materials in a project. The manager object is accessed via the definition manager.

```
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.

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	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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><material_name></td><td>string</td><td>Name of the project material</td></tr><tr><td></td><td>Boolean</td><td>True or False. If you use False or only a single argument, then GetProperties returns only the properties that are different from the defaults.</td></tr></table>
Return Value	Array of material data

Python Syntax	GetProperties (<materialname>)
Python Example	`oMaterialManager.GetProperties("Gold", True)` `oMaterialManager.GetProperties("vacuum")`

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

IsUsed [material manager]

Checks if a project material is in use.

UI Access	N/A
Parameters	Name: *<MaterialName>*; Type: String; Description: 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")`

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()`

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

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

IsUsed

Remove

RemoveUnused

Add [model manager]

Adds a model.

UI Access	**Tools > Edit Configured Libraries > Models > Add Model**		
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.

	<modelName>	String	Name of the model being added
	<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	Name of project and file path
	<PinDefInfo>	Array	Optional, defines a model pins; multiple <PinDefInfo> arrays can be included; contents of the <PinDefInfo> array are detailed below.
	<GraphicsDataInfo>	Array	Optional, defines graphics. The array contains the following: ["NAME:Graphics", # one or more of the following <RectInfo>, <CircleInfo>, <ArcInfo>, <LineInfo>, <PolygonInfo>, <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
```


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

```
<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
<bool>, # True if number is to be shown

<PropDisplayMapInfo>:
["NAME:PropDisplayMap",
<PropDisplayInfo>,
<PropDisplayInfo>,...]

<PropDisplayInfo>:
<NameString>, [<DisplayTypeInfo>,
<DisplayLocationInfo>,
<int>, // optional, level number
<TextInfo>]
```


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

```
<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>:
"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
```


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:=", [<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
<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
```


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

```
<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>,
<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",`

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

```
    "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]
    ]
  ])
```

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

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

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
	<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>,

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.
	<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.		

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, "",`

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

```
      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=***"]]
    ]
  ],
  ["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
```


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

```
<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>,
<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
```


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

```
# 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>]
```

<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
```


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

```
<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
<real> # y position
```

<PolygonInfo>:

```
"Polygon:=", [<real>, # line width
<int>, # fill pattern
<int>, # color
<PointInfo>, # must specify at least 3 points
<PointInfo>...]
```


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

<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
# 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
```


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]

Exports model(s) to a library.

UI Access	**Tools** > **Edit Configured Libraries** > **Models** > **Export to Library**		
Parameters	Name	Type	Description
	<*NameArray*>	Array	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.
	<*LibraryLocation*>	String	Location of the library specified with <LibraryName>; options are "`Project`", "`PersonalLib`", or "`UserLib`".
Return Value	None		

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

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	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
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

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><*ModelName*></td><td>String</td><td>Specifies the composite name of the model to remove</td></tr><tr><td><*IsProjectModel*></td><td>Boolean</td><td>True if it is a project model</td></tr><tr><td><*LibraryName*></td><td>String</td><td>Name of the library if it is not a project model</td></tr><tr><td><*LibraryLocation*></td><td>String</td><td>Location of the library in <LibraryName>; one of "Project", "PersonalLib", or "UserLib"</td></tr></table>
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.

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.

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 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()`

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	Type	Description

	<AddScriptArray>	Array	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.

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><*OriginalName*></td><td>String</td><td>Name of the script to be edited.</td></tr><tr><td><*EditScriptArray*></td><td>Array</td><td>Structured array.
	`["NAME:<string script name>",`
	`"ScriptLang:=", <string script language>`
	`"ScriptText:=, <string text of script>]`</td></tr></table>

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ScriptArray*></td><td>Array</td><td>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</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	ExportScript(<*ScriptArray*>,<*LibraryLocation*>)
Python Example	`oProject.ExportScript(["NAME:mylib", "myscript"], "PersonalLib")`

ModifyLibraries

Configures libraries accessed from the **Tools** menu.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*DesignName*></td><td>String</td><td>Name of specified design.</td></tr><tr><td><*ConfigLibArray*></td><td>Array</td><td>["NAME:<LibraryType>,<ConfiguredLib>,<ConfiguredLib>,…] <ConfiguredLib> # blank to leave unchanged <DefinitionType> ["<libraryname >","<libraryname>",…]</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	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		

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	RemoveScript (<*ScriptName*>,<*IsProjectScript*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveScript(` `"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

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

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

<ModifiedOnInfo>:

An integer that corresponds to the number of seconds that have elapsed

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

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

<PinType>:

<string> // "N" : normal pin

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

// "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

<TextInfo>)

<NameString>:

<string> // PropertyName:=, where PropertyName is the name of

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

// 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

<GraphicsDataInfo>:

Array("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:=", 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

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

<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

<int>, // line pattern

<int>, // color

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

<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>,

<bool>, // is plotter font

<string>, // font name

<int>, // color

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

<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

<ImageData>:
<string>, // file path

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 : 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

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.

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>Not available</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><SymbolName></td><td>String</td><td>Name of symbol to be edited</td></tr>
<tr><td><SymbolPropertiesArray></td><td>Array</td><td>Array of symbol properties</td></tr>
<tr><td><NAME></td><td>String</td><td>New simple name for the symbol</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><PinDefInfo></td><td>Array</td><td>Optional, it defines pin location, type, and color. Multiple <PinDefInfo> parameters can be included. See below for more information.</td></tr>
<tr><td><GraphicsDataInfo></td><td>Array</td><td>Optional; it defines graphics using one or more of the following:
<RectInfo>,
<CircleInfo>,
<ArcInfo>,
<LineInfo>,
<PolygonInfo>,
<TextInfo>,
<ImageInfo>
See below for more details.</td></tr>
<tr><td><PropDisplayMapInfo></td><td>Array</td><td>Optional, it defines display properties. See below for more information.</td></tr>
<tr><td><ListOfComponentNames></td><td>Array</td><td>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.</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.

	<*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.
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",T-
    rue],
    [
      "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",
```


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

```
        "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-
        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
```


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

```
# "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>:

```
["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
```


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

```
        # 3 : Bottom
        # 4 : Center
        # 5 : Custom placement
```

<GraphicsDataInfo>

```
["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>:

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

```
"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
<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>,
```


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> – 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
# 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
```


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

<CompName> – Name of the component defined as a string

<TermAttributes> – An array that includes the following:

```
<Terminal Name> # defined as a string
<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>*)

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()`
	`oSymbolManager = oDefinitionManager.GetManager("Symbol")`
	`oSymbolManager.Export(["NAME Nexxim Circuit Ele-`
	`ments\Distributed\Distributed:bendo:mylib", "mySymbol"], "PersonalLib")`

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	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*>)
Python Example	`oSymbolManager.Remove("Nexxim Circuit Elements\Distributed\Distributed:bendo", 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.

RemoveUnused [symbol manager]

Removes symbols 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 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()`

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 - 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.

26 - Example Scripts

This section contains IronPython example scripts.

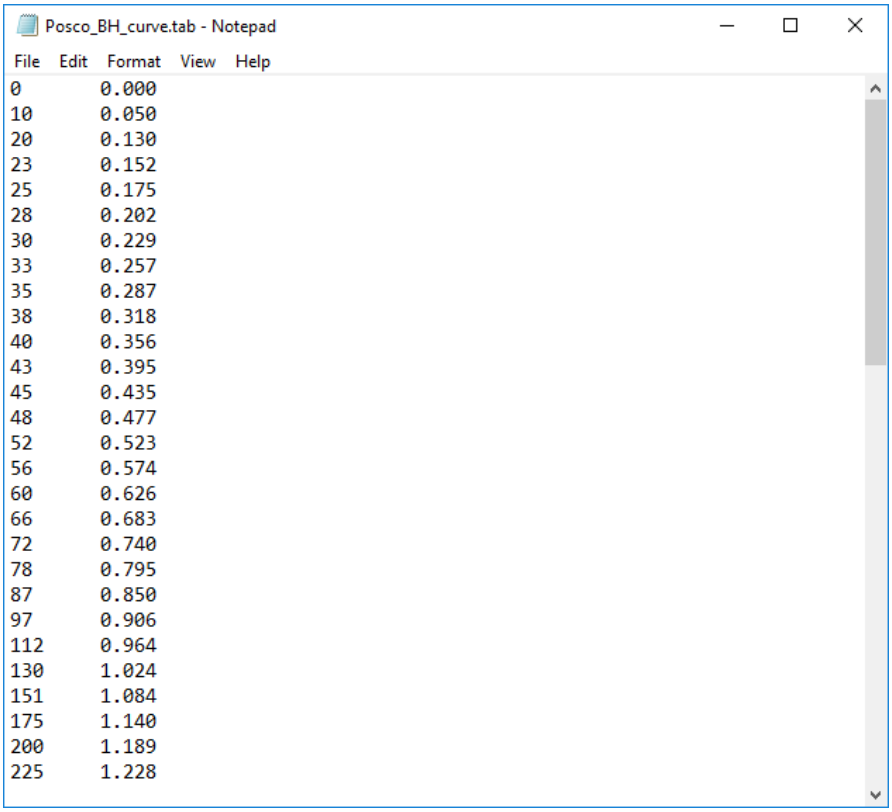
IronPython Example Scripts

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
  ])
```

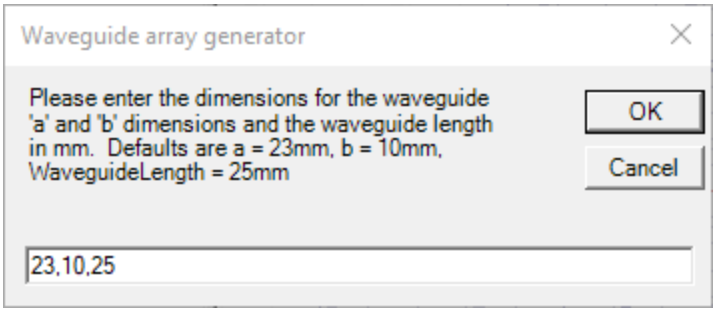

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

HFSS Waveguide Array Python Script

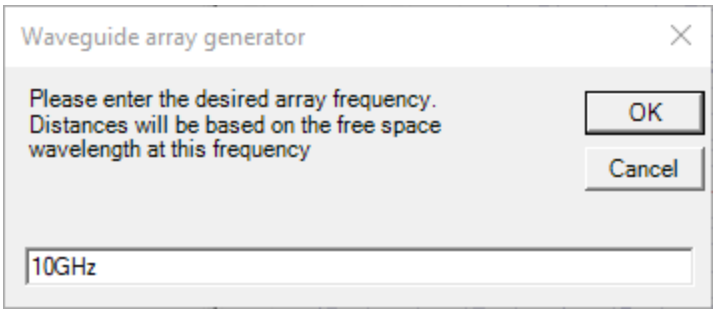
This sample Python script creates an HFSS Waveguide array. The script includes comment lines, which are preceded by an apostrophe ('), that offer explanations for each subsequent line or lines. The script includes examples of creating a 3D model, boundaries and excitation, solution setup and sweep, as well as reports.

You must insert an HFSS project before running the script. Running the script causes a series of input dialogs to appear that lets you set parameters.

The first dialog asks for input for a and be dimensions and the waveguide length.

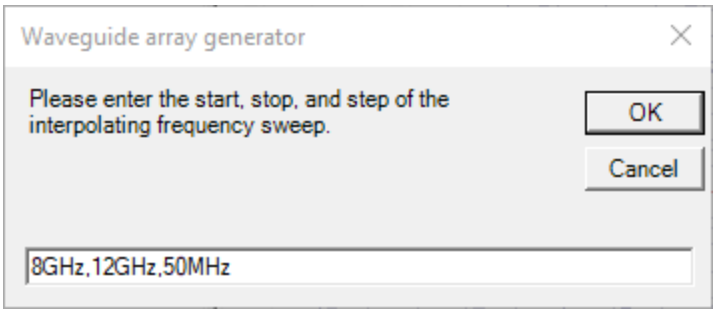


The next asks for the array frequency.

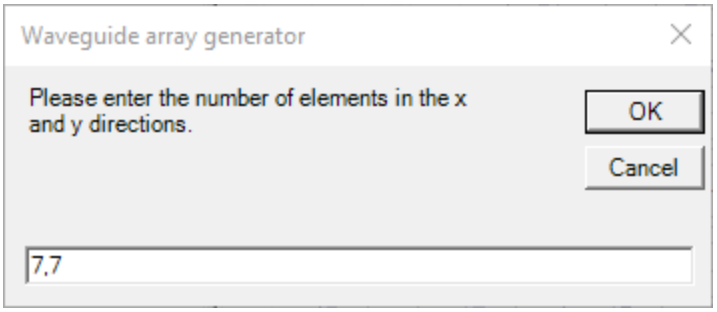


The next asks for start, stop and step values for in an interpolating frequency sweep.

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

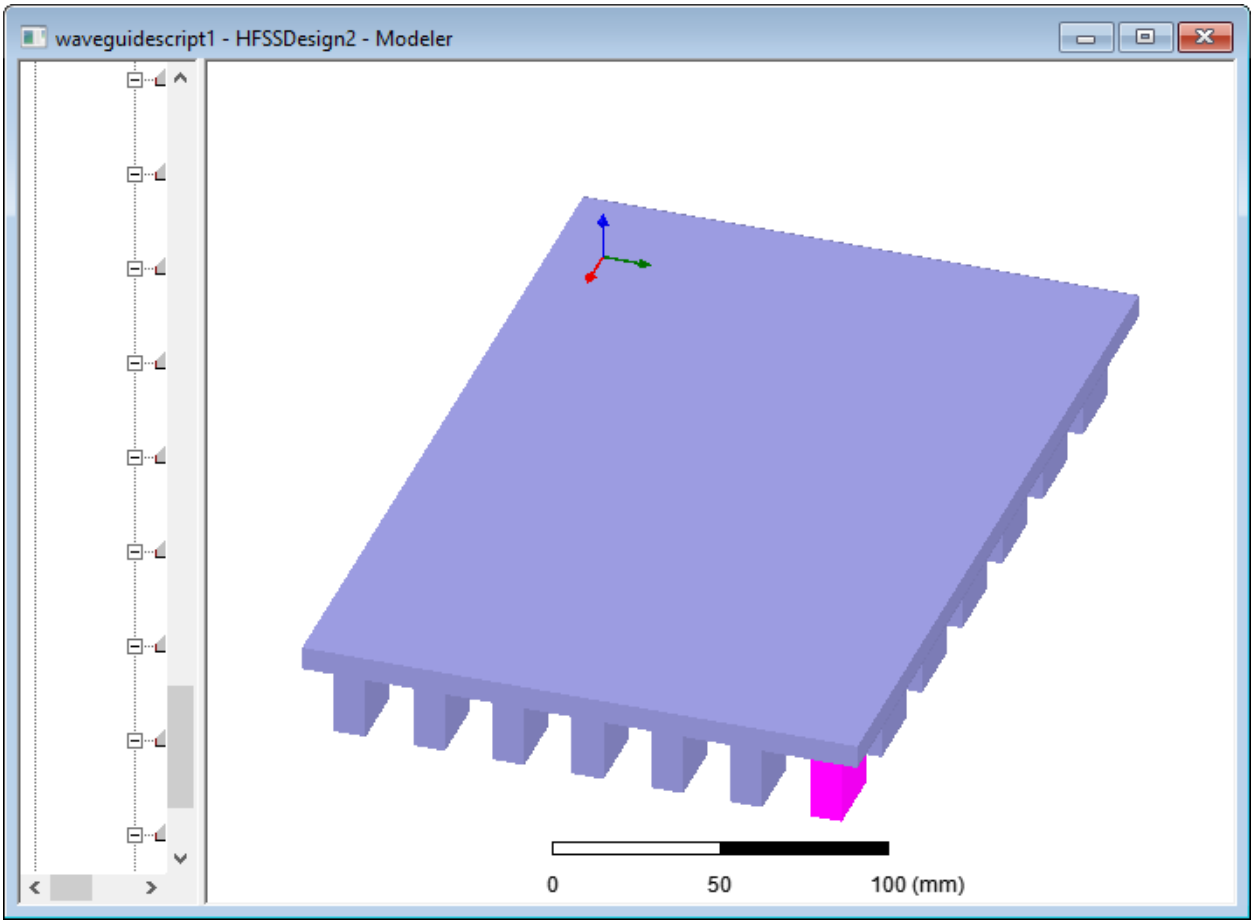


The last one asks for the array definition.



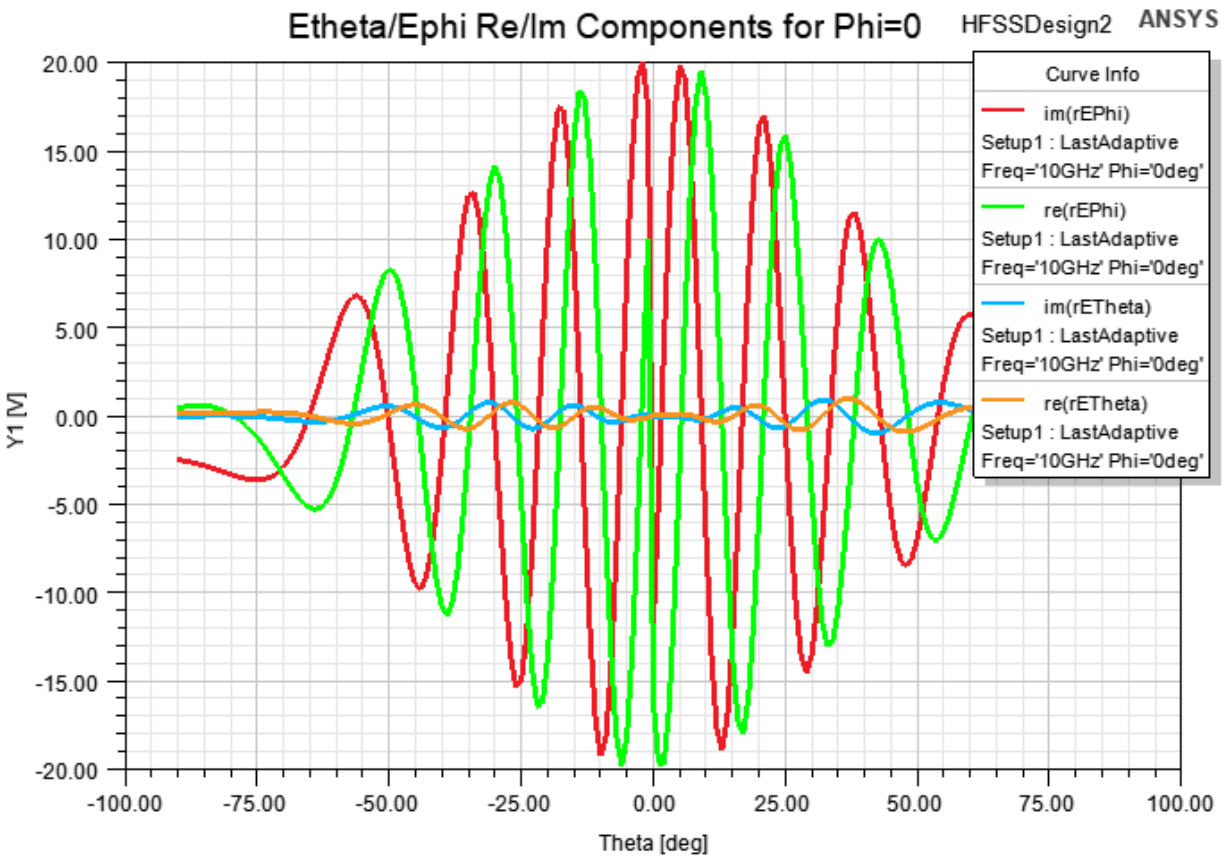
The following figure shows the waveguide array generated by the defaults.

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

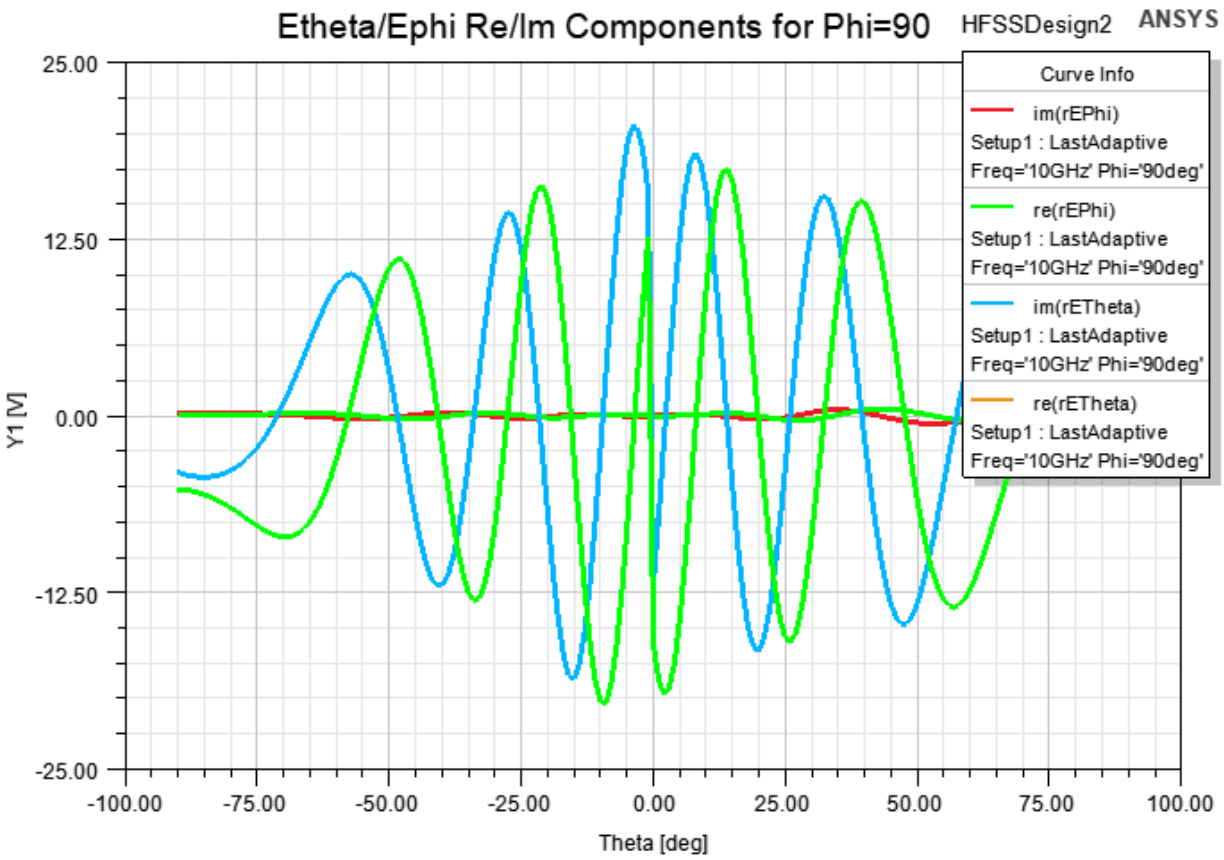


After running the simulation, the plots defined by the script shows the following results.

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.



The waveguide script in Python follows.

```
import clr
import os
import sys
import time

clr.AddReference("System.Windows.Forms")

from Microsoft.Win32 import Registry
global AnsysStandalone, AnsysInstallDirectory, Desktop
desktopVersion = '2024.2'
installDirectory = Registry.GetValue("HKEY_LOCAL_MACHINE\Soft-
ware\Ansoft\ElectronicsDesktop\{}\Desktop".format(desktopVersion),
"InstallationDirectory", '')
sys.path.append(installDirectory)
clr.AddReference("Ansys.Ansoft.DesktopPlugin")
clr.AddReference("Ansys.Ansoft.CoreCOMScripting")
clr.AddReference("Ansys.Ansoft.PluginCoreDotNet")
sys.path.append(os.path.join(installDirectory,
```


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

```
'PythonFiles','DesktopPlugin'))
import ScriptEnv

ScriptEnv.Initialize('Ansoft.ElectronicsDesktop.{}'.format(desktopVer-
sion))
Desktop = sys.modules['__main__'].oDesktop
Desktop.RestoreWindow()

clr.AddReferenceByPartialName("Microsoft.VisualBasic")
from Microsoft.VisualBasic.Constants import
vbOKOnly,vbOKCancel,vbAbortRetryIgnore,vbYesNoCancel,vbYesNo,vbRetryC-
ancel
from Microsoft.VisualBasic.Constants import
vbOK,vbCancel,vbAbort,vbRetry,vbIgnore,vbYes,vbNo
from Microsoft.VisualBasic.Interaction import InputBox, MsgBox

oProject = oDesktop.NewProject()
oProject.InsertDesign("HFSS", "HFSSDesign1", "HFSS Terminal Network",
"")

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

# Ask for dimensions for waveguide dimensions
# ----------------------------------------------------------

dim = InputBox("Please enter the dimensions for the waveguide 'a' and
b' dimensions " + "and the waveguide length in mm. Defaults are a =
23mm, b = 10mm, WaveguideLength = 25mm", "Waveguide array generator",
"23,10,25")
Dimensions = dim.split(',')

a_str = Dimensions[0] + "mm"
b_str = Dimensions[1] + "mm"
b_over2 = float(Dimensions[1])/2

WaveguideLength_str = Dimensions[2] + "mm"

#Ask for the frequency of operation
#---------------------------------
Frequency = InputBox("Please enter the desired array frequency.
Distances will " + "be based on the free space wavelength at this fre-
quency", "Waveguide array generator", "10GHz")

#Ask for the start, stop, and step of the interpolating frequency
sweep
#-----------------------------------------------------------------
--
inputText = InputBox("Please enter the start, stop, and step of the
interpolating " + "frequency sweep.", "Waveguide array generator",
"8GHz,12GHz,50MHz")
StartFrequency, StopFrequency, StepFrequency = inputText.split(',')
```


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

```
#Ask for the number of elements in the x and y directions
#-----------------------------------------------------
inputText = InputBox("Please enter the number of elements in the x
and y directions.", "Waveguide array generator", "7,7")
numX, numY = [float(elem) for elem in inputText.split(',')]
TotalElements = numX*numY

# Make variables and set them equal to the values from the user input
# ------------------------------------------------------------------
oDesign.ChangeProperty(

  [

    "NAME:AllTabs",

    [

    "NAME:LocalVariableTab",

    [

    "NAME:PropServers",
    "LocalVariables"
    ],
    [
    "NAME:NewProps",
    [

      "NAME:a", "PropType:=", "VariableProp", "UserDef:=", True,
      "Value:=", a_str

    ],
    [

      "NAME:b", "PropType:=", "VariableProp", "UserDef:=", True,
      "Value:=", b_str

    ],
    [

      "NAME:NumX", "PropType:=", "VariableProp", "UserDef:=", True,
      "Value:=", numX

    ],
    [

      "NAME:NumY", "PropType:=", "VariableProp", "UserDef:=", True,
      "Value:=", numY

    ],
    [

      "NAME:WaveguideLength", "PropType:=", "VariableProp", "User-
      Def:=", True, "Value:=", WaveguideLength_str
```


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

```
    ],
    [
      "NAME:Frequency","PropType:=", "VariableProp", "UserDef:=",
      True, "Value:=", Frequency
    ],
    [
      "NAME:Lambda", "PropType:=", "VariableProp", "UserDef:=", True,
      "Value:=", "c0/" + Frequency
    ],
    [
      "NAME:RadBoundDist", "PropType:=", "VariableProp", "UserDef:=",
      True, "Value:=", "Lambda/4"
    ]
    ]
    ]
    ])

#Create the radiation box
#------------------------
oEditor.CreateBox(
  [
    "NAME:BoxParameters",
    "XPosition:=" , "-a/2-RadBoundDist",
    "YPosition:=" , "-b/2-RadBoundDist",
    "ZPosition:=" , "0mm",
    "XSize:=" , "NumX*a+(NumX-1)*Lambda/2+2*RadBoundDist",
    "YSize:=" , "NumY*b+(NumY-1)*Lambda/2+2*RadBoundDist",
    "ZSize:=" , "RadBoundDist"
  ],
  [
    "NAME:Attributes",
    "Name:=" , "RadiationBox",
    "Flags:=" , "",
    "Color:=" , "(132 132 193)",
    "Transparency:=" , 0.8,
    "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.

```
  ])
oEditor.FitAll() # Zoom out
#Create first element
#--------------------
oEditor.CreateBox(
  [
    "NAME:BoxParameters",
    "XPosition:=" , "-a/2",
    "YPosition:=" , "-b/2",
    "ZPosition:=" , "0mm",
    "XSize:=" , "a",
    "YSize:=" , "b",
    "ZSize:=" , "-WaveguideLength"
  ],
  [
    "NAME:Attributes",
    "Name:=" , "Element1",
    "Flags:=" , "",
    "Color:=" , "(132 132 193)",
    "Transparency:=" , 0.8,
    "PartCoordinateSystem:=", "Global",
    "UDMId:=" , "",
    "MaterialValue:=" , "\"vacuum\"",
    "SurfaceMaterialValue:=", "\"\"",
    "SolveInside:=" , True,
    "IsMaterialEditable:=" , True,
    "UseMaterialAppearance:=", False,
    "IsLightweight:=" , False
  ])
# Define the port
# ---------------
# Get the numeric value of half of the b dimension of the port.
# The values fed in to the "Start" and "End" arrays cannot have math-
ematical operators. For example, if HFSS
# has a variable 'b', and a desired coordinate is 'b/2', that value
cannot be entered here as "b/2". The number
# has to be computed explicitly in script and entered as a string
such as "-200mm".
oModule = oDesign.GetModule("BoundarySetup")
# get bottom face ID
element1FaceID = oEditor.GetFaceByPosition(
  [
```


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",
    "BodyName:=", "Element1",
    "XPosition:=", "-a/2",
    "YPosition:=", "-b/2",
    "ZPosition:=", "-WaveguideLength"

    ])

oModule.AssignWavePort(

  [

    "NAME:WavePort1",
    "Faces:=" , [element1FaceID],
    "NumModes:=" , 1,
    "RenormalizeAllTerminals:=", True,
    "UseLineModeAlignment:=", False,
    "DoDeembed:=" , False,

  [

    "NAME:Modes",
    [
    "NAME:Mode1",
    "ModeNum:=" , 1,
    "UseIntLine:=" , True,
    [
    "NAME:IntLine", "Start:=" , ["0mm", "-" +str(b_over2) + "mm", "-"
    + WaveguideLength_str], "End:=" , ["0mm", str(b_over2) + "mm", "-
    " + WaveguideLength_str]
    ],
    "AlignmentGroup:=" , 0, "CharImp:=" , "Zpi"
    ]
    ],

  "ShowReporterFilter:=" , False, "ReporterFilter:=" , [True],
  "UseAnalyticAlignment:=", False

  ])

#Set the radiation boundary
#--------------------------
# Get face IDs for further assignment
top_face_id = oEditor.GetFaceByPosition(

  [

    "NAME:FaceParameters",
    "BodyName:=", "RadiationBox",
    "XPosition:=", "0mm",
    #"YPosition:=", "NumY*b-b/2+(NumY-1)*Lambda/2+RadBoundDist",
    "YPosition:=", "0mm",
    "ZPosition:=", "0mm"
```


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

```
  ])
#AddInfoMessage("top_face_id :")
#AddInfoMessage(str(top_face_id))
faceIDs = [int(elem) for elem in oEditor.GetFaceIDs("RadiationBox")
if elem != str(top_face_id)]
#AddInfoMessage("faceIDs :")
#AddInfoMessage(str(faceIDs))
oModule.AssignRadiation(
  [
    "NAME:Radiation",
    "Faces:=" , faceIDs,
    "IsFssReference:=" , False,
    "IsForPML:=" , False
  ])


# Copy and paste elements/ports into a rectangular array.
# Duplicate boundaries with geometry" must be turned on under Tools-
>Options->HFSS Options
# ----------------------------------------------------------------
----------------------
ElementNum = 1
for i in range(1, int(numX)+1):
  for j in range(1, int(numY)+1):
    if ElementNum == 1:
      pass
    elif ElementNum <= numY: #If in the first column, only
      oEditor.Copy(["NAME:Selections", "Selections:=", "Element1"])
      time.sleep(0.1) # added to give copy command time to finish
      copying object before paste
      oEditor.Paste()
      oEditor.Move(["NAME:Selections", "Selections:=", "Element" +
      str(ElementNum)],
      [
        "NAME:TranslateParameters",
        "CoordinateSystemID:=", -1,
        "TranslateVectorX:=", "0mm",
        "TranslateVectorY:=", str(j-1) + "*(b+Lambda/2)",
        "TranslateVectorZ:=", "0mm"
      ])
```


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

```
    elif ElementNum > numY:

      oEditor.Copy(["NAME:Selections", "Selections:=", "Element1"])
      time.sleep(0.1) # added to give copy command time to finish
      copying object before paste
      oEditor.Paste()
      oEditor.Move(["NAME:Selections", "Selections:=", "Element" +
      str(ElementNum )],
      [

        "NAME:TranslateParameters",
        "CoordinateSystemID:=", -1,
        "TranslateVectorX:=", str(i-1) + "*(a+Lambda/2)",
        "TranslateVectorY:=", str(j-1) + "*(b+Lambda/2)",
        "TranslateVectorZ:=", "0mm"

    ])

  ElementNum += 1
  #AddInfoMessage(str(ElementNum))

#Create the setup and interpolating sweep
#----------------------------------------
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("HfssDriven",

  [

    "NAME:Setup1",
    "AdaptMultipleFreqs:=" , False,
    "Frequency:=" , Frequency,
    "MaxDeltaS:=" , 0.02,
    "PortsOnly:=" , False,
    "UseMatrixConv:=" , False,
    "MaximumPasses:=" , 15,
    "MinimumPasses:=" , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:=" , 50,
    "IsEnabled:=" , True,
    "BasisOrder:=" , 1,
    "DoLambdaRefine:=" , True,
    "DoMaterialLambda:=" , True,
    "SetLambdaTarget:=" , False,
    "Target:=" , 0.3333,
    "UseMaxTetIncrease:=" , False,
    "PortAccuracy:=" , 2,
    "UseABCOnPort:=" , False,
    "SetPortMinMaxTri:=" , False,
    "UseDomains:=" , False,
    "UseIterativeSolver:=" , False,
    "SaveRadFieldsOnly:=" , False,
    "SaveAnyFields:=" , True,
```


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

```
    "IESolverType:=" , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True

  ])
oModule.InsertFrequencySweep("Setup1",
  [
    "NAME:InterpolatingSweep",
    "IsEnabled:=" , True,
    "RangeType:=" , "LinearStep",
    "RangeStart:=" , StartFrequency,
    "RangeEnd:=" , StopFrequency,
    "RangeStep:=" , StepFrequency,
    "Type:=" , "Interpolating",
    "SaveFields:=" , False,
    "InterpTolerance:=" , 0.5,
    "InterpMaxSolns:=" , 50,
    "InterpMinSolns:=" , 0,
    "InterpMinSubranges:=" , 1,
    "ExtrapToDC:=" , False,
    "InterpUseS:=" , True,
    "InterpUsePortImped:=" , False,
    "InterpUsePropConst:=" , True,
    "UseDerivativeConvergence:=", False,
    "InterpDerivTolerance:=", 0.2,
    "UseFullBasis:=" , True,
    "EnforcePassivity:=" , True,
    "PassivityErrorTolerance:=", 0.0001

  ])
# pattern calculations
#-------------------------------------------------------------------------
------------------------
oEditor.CreateRelativeCS(
  [
    "NAME:RelativeCSParameters",
    "Mode:=" , "Axis/Position",
    "OriginX:=" , "-a/2+NumX*a/2+(NumX-1)*Lambda/4",
    "OriginY:=" , "-b/2+NumY*b/2+(NumY-1)*Lambda/4",
    "OriginZ:=" , "0mm",
    "XAxisXvec:=" , "1mm",
    "XAxisYvec:=" , "0mm",
    "XAxisZvec:=" , "0mm",
    "YAxisXvec:=" , "0mm",
    "YAxisYvec:=" , "1mm",
    "YAxisZvec:=" , "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:Attributes",
    "Name:=" , "RelativeCS1"

  ])

#Create an infinite sphere with fine theta resolution and phi cuts at
0 and 90 degrees
#-------------------------------------------------------------------
-----------------
oModule = oDesign.GetModule("RadField")
oModule.InsertFarFieldSphereSetup(

  [

    "NAME:Infinite Sphere1",
    "UseCustomRadiationSurface:=", False,
    "ThetaStart:=" , "-90deg",
    "ThetaStop:=" , "90deg",
    "ThetaStep:=" , "1deg",
    "PhiStart:=" , "0deg",
    "PhiStop:=" , "90deg",
    "PhiStep:=" , "90deg",
    "UseLocalCS:=" , True,
    "CoordSystem:=" , "RelativeCS1"

  ])

# run the simulation
oDesign.AnalyzeAll()

#Create output plots for Ephi/Etheta real and imaginary components
for Phi = 0 and separately for Phi = 90
#-------------------------------------------------------------------
-----------------------------------

# create report for pho=0
oModule = oDesign.GetModule("ReportSetup")
oModule.CreateReport("Etheta/Ephi Re/Im Components for Phi=0", "Far
Fields", "Rectangular Plot", "Setup1 : LastAdaptive",

  [

    "Context:=" , "Infinite Sphere1"

  ],
  [

    "Theta:=" , ["All"],
    "Phi:=" , ["0deg"],
    "Freq:=" , ["10GHz"],
    "a:=" , ["Nominal"],
```


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

```
    "b:=" , ["Nominal"],
    "NumX:=" , ["Nominal"],
    "NumY:=" , ["Nominal"],
    "WaveguideLength:=" , ["Nominal"],
    "Frequency:=" , ["Nominal"]

  ],
  [

    "X Component:=" , "Theta",
    "Y Component:=" , ["re(rEPhi)","im(rEPhi)","re(rETheta)","im
    (rETheta)"]

  ])

# create report for pho=00
oModule.CreateReport("Etheta/Ephi Re/Im Components for Phi=90", "Far
Fields", "Rectangular Plot", "Setup1 : LastAdaptive",

  [

    "Context:=" , "Infinite Sphere1"

  ],
  [

    "Theta:=" , ["All"],
    "Phi:=" , ["90deg"],
    "Freq:=" , ["10GHz"],
    "a:=" , ["Nominal"],
    "b:=" , ["Nominal"],
    "NumX:=" , ["Nominal"],
    "NumY:=" , ["Nominal"],
    "WaveguideLength:=" , ["Nominal"],
    "Frequency:=" , ["Nominal"]

  ],
  [

    "X Component:=" , "Theta",
    "Y Component:=" , ["re(rEPhi)","im(rEPhi)","re(rETheta)","im
    (rETheta)"]

  ])
```


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.

Index

#

3D Modeler editor commands 8-1
- AlignFaces 8-100
- AssignMaterial 8-101
- AssignSurfaceMaterial 8-199
- Chamfer 8-103
- CleanUpModel 8-104
- CloseAllWindows 8-202
- Connect 8-104
- Copy 8-84
- CoverLines 8-105
- CoverSurfaces 8-106
- CreateBondwire 8-6
- CreateBox 8-9
- CreateCircle 8-11
- CreateCone 8-13
- CreateCutplane 8-15
- CreateCylinder 8-17
- CreateEllipse 8-19
- CreateEntityList 8-106, 8-113
- CreateEquationCurve 8-21
- CreateEquationSurface 8-24
- CreateFaceCS 8-108, 8-114
- CreateGroup 8-111
- CreateHelix 8-26
- CreateObjectFromdges 8-118
- CreateObjectFromFace 8-120
- CreateObjectFromFaces 8-121
- CreatePoint 8-28
- CreatePolyline 8-29
- CreateRectangle 8-32
- CreateRegularPolygon 8-34
- CreateRegularPolyhedron 8-36
- CreateRelativeCS 8-122
- CreateSpiral 8-38
- CreateTorus 8-40
- CreateUserDefinedModel 8-42
- CreateUserDefinedPart 8-49
- Defeature 8-203
- Delete 8-203
- DeleteEmptyGroups 8-124
- DeleteLastOperation 8-125
- DeleteOperation 8-125
- DeletePolylinePoint 8-85
- DetachEdges 8-127
- DetachFaces 8-128
- DuplicateAlongLine 8-86
- DuplicateAroundAxi 8-88
- DuplicateMirror 8-89
- EditEntityList 8-129, 8-134

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

EditFaceCS 8-130

EditObjectCS 8-135

EditPolyline 8-60

EditRelativeCS 8-140

Export 8-141

ExportModelImagetoFile 8-143, 10-47

Fillet 8-147

FlattenGroup 8-148

GenerateAllUserDefinedModels 8-205

GenerateHistory 8-149

GenerateUserDefinedModel 8-205

Get3DComponentParameters 8-65

Get3DComponentPartNames 8-66

GetBodyNamesByPosition 8-207

GetEdgeByPosition 8-214

GetEdgeIDFromNameForFirstOperation 8-215

GetEdgeIDsfromFace 8-216

GetEdgeIDsfromObject 8-216

GetEdgeLength 8-217

GetEdgePositionAtNormalizedParameter 7-37

GetEntityListIDByName 8-217, 8-218

GetFaceArea 8-219

GetFaceByPosition 8-220

GetFaceCenter 8-219

GetFaceIDFromNameForFirstOperation 8-221

GetFaceIDs 8-222

GetFaceIDsOfSheet 8-222

GetGeometryModelerMode 8-223

GetGroupSubmodelNames 8-223

GetModelBoundingBox 8-225

GetPartsForUserDefinedModel 8-233

GetPoints 8-233

GetProperties 2-39, 6-34, 7-46

GetPropertyValue 2-40, 6-35, 7-47, 8-234, 10-68, 24-59

GetPropEvaluatedValue 8-234

GetPropSIValue 8-237

GetRelativeCoordinateSystems 8-239

GetUser Position 8-240

GetVertexIDFromNameForFirstOperation 8-241

GetVertexIDsFromEdge 8-242

GetVertexIDsFromFace 8-242

GetVertexIDsFromObject 8-243

GetVertexPosition 8-243

GetWireBodyNames 8-244

HealObject 8-150

Import 8-152

ImportDXF 8-155

ImportFromClipboard 8-157

ImportGDSII 8-158

Imprint 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.

ImprintProjection 8-161
Insert3DComponent 8-66
InsertComponent 8-68
InsertNativeComponent 8-69
Intersect 8-163
Mirror 8-92
Move 8-93
MoveCSToEnd 8-164
MoveEntityToGroup 8-164
MoveFaces 8-165
OffsetFaces 8-94
PageSetup 8-244
ProjectSheet 8-167
RemoveBadEdges 8-246
RemoveBadFaces 8-246
RemoveBadVertices 8-247
RenamePart 8-247
Replacewith3DComponent 8-168
Rotate 8-96
Scale 8-97
Section 8-170
SeparateBody 8-172
SetModelUnits 8-172
SetPropValue Modeler 8-248
SetTopDownViewDirectionForActiveView 8-253
SetTopDownViewDirectionForAllViews 8-254
SetWCS 8-173
Simplify 8-175
Split 8-177
Stitch 8-179
Subtract 8-180
SweepAlongPath 8-75
SweepAlongVector 8-76
SweepAroundAxis 8-78
SweepFacesAlongNormal 8-80, 8-80, 8-181, 8-181
SweepFacesAlongNormalWithAttributes 8-81
ThickenSheet 8-182
UncoverFaces 8-183
Ungroup 8-186
Unite 8-184
UpdateComponentDefinition 8-83
UpdatePriorityList 8-254
UpgradeVersion 8-255
Validate3DComponent 8-256
WrapSheet 8-186
WriteHistoryTreeLayoutForTest 8-257
3D Modeler editor object commands
GetNumObjects 8-226
GetObjectIDByName 8-226
GetObjectName 8-227
GetObjectNameByEdgeID 8-227
GetObjectNameByFaceID 8-228
GetObjectNameByID 8-228

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

GetObjectNameByVertexID 8-229
GetObjectShapeType 8-230
GetObjectVolume 8-231
GetObjPath 8-232

A

Add 24-13, 24-19, 24-74, 24-96
AddAllEyeMeasurements 10-4
AddAssignmentToBoundary 11-2
AddCableToBundle 8-259
AddCartesianLimitLine 10-4
AddCartesianLimitLineFromCurve 10-5
AddCartesianLimitLineFromEquation 10-7
AddCartesianXMarker 10-8
AddCartesianYMarker 10-9
AddCartesianYMarkerToStack 10-9
AddDataset 6-4, 7-4, 24-5
AddDefinitionFromBlock 8-190, 24-7
AddDefinitionFromLibFile 8-192
AddDeltaMarker 10-10
AddDiffPair 23-3
AddMarker 10-11
AddMarker[Fields Reporter] 18-1
AddMarkerToPlot 18-2
AddMaterial 6-6, 24-9
AddMessage 4-6
AddModelingProperties 7-6
AddNote 10-12
AddTraceCharacteristics 10-13
AddTraces 10-14
AlignFaces 8-100
Analysis module commands
AnalyzeAll 7-6
AnalyzeAllNominal 7-7
CopySetup 15-2, 16-6, 16-39
DeleteSetups 15-3
EditSetup 15-3, 16-8, 16-41
GetSetupCount 15-15
GetSetups 15-15
PasteSetup 15-33
RenameSetup 15-33
RevertAllToInitial 15-33
RevertSetupToInitial 15-34
RevertSetupToInitialConditions 15-35
RevertToInitialCondition 7-60
Analysis Setup Module Script Commands 15-1
Ansoft Application Object commands 3-1
GetAppDesktop 3-2
ApplyMeshOps 7-8, 7-34
ApplyReportTemplate 10-16
AssignAdiabaticPlateBoundary 11-18
AssignBlockBoundary 11-12
AssignBlowerBoundary_1 11-29
AssignConductingPlateBoundary 11-19

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

AssignEMLoss 11-27
AssignFaceMonitor 13-1
AssignGrilleBoundary 11-14
AssignHeatExchangerBoundary 11-30
AssignInitialTemperature 11-32
AssignMaterial 8-101
AssignMeshOperation 14-5, 14-6
AssignMeshRegion 14-3
AssignNetworkBoundary 11-15
AssignOpeningBoundary 11-16
AssignPointMonitor 13-2
AssignRecircBoundary 11-17
AssignResistanceBoundary 11-21
AssignSolarLoading 11-31
AssignSourceBoundary 11-24
AssignStationaryWallBoundary 11-25
AssignSurfaceMaterial 8-199
AssignSymmetryWallBoundary 11-27
AssignVirtualMeshRegion 14-7

B

Boundary and Excitation Module Script Commands 11-1, 12-1
boundary conditions script commands 11-1
Boundary/Excitation module commands
AddAssignmentToBoundary 11-2
DeleteAllBoundaries 11-3
DeleteBoundaries 11-3
GetAssignment 11-7
GetBoundaries 11-6
GetBoundariesofType 11-6
GetDefaultBaseName 11-7
GetNumBoundaries 11-8
GetNumBoundariesofType 11-8
ReassignBoundaries 11-2, 11-8
ReassignBoundary 11-8
RemoveAssignmentFromBoundary 11-9
RenameBoundary 11-10
ReprioritizeBoundaries 11-10
SetDefaultBaseName 11-11
BreakUDMConnection 8-201
BringToFront 24-105

C

Cable Modeling Commands 8-258
Cable Modeling commands
AddCableToBundle 8-259
CreateCableBundle 8-260
CreateCableHarness 8-261
CreateClockSource 8-263
CreatePWLSource 8-264
CreateStraightWireCable 8-266, 8-267
ExportCableLibrary 8-268
ImportCableLibrary 8-269

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

RemoveCables 8-269
UpdateCableHarness 8-270
CalcOp 18-3, 19-3
CalcRead (deprecated) 19-3
CalcStack 18-4, 19-5
CalculatorRead 18-5, 19-4
CalculatorWrite 18-5, 19-5
Cascade 23-4
Chamfer 8-103
ChangeGeomSettings 18-6, 19-6
ClcEval 18-7, 19-7
ClcMaterial 18-7, 19-7
ClcMaterialValue 18-8, 19-8
CleanUpModel 8-104
ClearAllMarkers 10-17
ClearAllMarkers[Fields Reporter] 18-8
ClearAllNamedExpr 18-9, 19-9
ClearAllTraceCharacteristics 10-17
ClearDiffPairs 23-5
ClearMessages 4-7, 6-10
Clone 23-6
CloneReportsFromDatasetSolution 10-18
Close 6-10, 23-7
CloseAllWindows 4-10, 8-202
CloseProject 4-11
CloseProjectNoForce 4-12
Combine 23-8
Connect 8-104
ConstructVariationString 7-8
conventions
scripting help 1-1
ConvertToDynamic 24-80
ConvertToParametric 24-81
Copy 8-84
CopyDesign 6-11
CopyItemCommand 7-9
CopyNamedExprToStack 18-9, 19-9
CopyPlotSettings 10-19
CopyReportDefinition 10-19
CopyReportsData 10-20
Copyright and Trademark Information 2
CopySetup
Analysis module command 15-2, 16-6, 16-39
Optimetrics module command 15-2, 16-6, 16-39
CopyTraceDefinitions 10-20
CopyTracesData 10-21
Core Global Script Context Commands 25-1
Count 4-12
CoverLines 8-105
CoverSurfaces 8-106
CPython 1-44
CreateBondwire 8-6
CreateBox 8-9
CreateCableBundle 8-260
CreateCableHarness 8-261

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

CreateCircle 8-11
CreateClockSource 8-263
CreateCone 8-13
CreateCutplane 8-15
CreateCylinder 8-17
CreateEllipse 8-19
CreateEntityList 8-106, 8-113
CreateEquationCurve 8-21
CreateEquationSurface 8-24
CreateFaceCS 8-108, 8-114
CreateGroup 8-111
CreateHelix 8-26
CreateObjectFromEdges 8-118
CreateObjectFromFace 8-120
CreateObjectFromFaces 8-121
CreateOutputVariable 9-1
CreatePcbPowermap 7-9
CreatePoint 8-28
CreatePolyline 8-29
CreatePWLSource 8-264
CreateRectangle 8-32
CreateRegularPolygon 8-34
CreateRegularPolyhedron 8-36
CreateRelativeCS 8-122
CreateReport 7-10, 10-22
CreateReportFromTemplate 10-34
CreateReportofAllQuantities 10-35
CreateSpiral 8-38
CreateTorus 8-40
CreateUserDefinedModel 8-42
CreateUserDefinedPart 8-49
CreateUserDefinedSolution 22-1
CutDesign 6-12

D

dataset commands
- AddDataset 6-4, 7-4, 24-5
- DeleteDataset 6-12, 7-23, 24-27
- EditDataset 6-14, 24-40
- ExportDataset 6-23, 7-30, 24-54
- HasDataset 6-39
- ImportDataset 6-39, 7-53, 24-60

Deembed 23-9
DeembedBack 23-10
DeembedFront 23-11
Defeature 8-203
Definition Manager commands
- AddDefinitionFromBlock 8-190, 24-7
- AddDefinitionFromLibFile 8-192
- GetExtendedDefinitionObject 8-218
- GetProjectMaterialNames 24-58
- UpdateDefFromBlock 24-69

Definition Manager Script Commands 24-1
Delete 8-203
DeleteAllBoundaries 11-3
DeleteAllReports 10-37

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

DeleteBoundaries 11-3

DeleteDataset 6-12, 7-23, 24-27

DeleteDesign 6-13

DeleteEmptyGroups 8-124

DeleteFieldPlot 18-13

DeleteLastOperation 8-125

DeleteMarker 10-36

DeleteMarker[Fields Reporter] 18-14

DeleteNamedExpr 18-15, 19-10

DeleteOperation 8-125

DeleteOutputVariable 7-24, 7-25, 9-2

DeletePolylinePoint 8-85

DeleteProject 4-13

DeleteReport 10-37

DeleteSetups 15-3

Analysis module command 15-3

Optimetrics module command 16-6, 16-39

DeleteSolutionVariation 7-24, 17-2

DeleteTraceCharacteristics 10-38

Deletetraces 10-38

DeleteUneditablePlot 18-15

DeleteUserDefinedSolutions 22-2

Design object commands 10-58, 10-59, 10-60, 10-61, 10-61, 10-73

AddCartesianLimitLine 10-4

AddCartesianXMarker 10-8

AddCartesianYMarker 10-9

AddDeltaMarker 10-10

AddMarker 10-11

AddMarkerToPlot 18-2

AddNote 10-12

AddTraceCharacteristics 10-13

AddTraces 10-14

ApplyMeshOps 7-8, 7-34

CalculatorRead 18-5, 19-4

ClearAllMarkers 10-17

ClearAllTraceCharacteristics 10-17

CloseAllWindows 4-10

CopyReportDefinition 10-19

CreateOutputVariable 9-1

CreateReportFromTemplate 10-34

DeleteAllReports 10-37

DeleteOutputVariable 7-24, 7-25, 9-2

DeleteReport 10-37

DeleteTraces 10-38

DoesOutputVariableExist 9-3

EditDesignSettings 7-26

EditMarker 10-42

EditOutputVariable 9-4

ExportConvergence 7-29

ExportModelMeshToFile 8-146, 10-51

ExportPlot3DToFile 10-51

ExportPlotImageToFile 18-30

ExportPlotImageWithViewToFile 18-31

ExportProfile 7-32

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

ExportReport 10-52

ExportToFile 10-56

GeometryCheckAndAutofix 8-206

GetDisplayType 10-64

GetLibraryDirectory 4-24

GetMatchedObjectName 8-224

GetModelUnits 8-225

GetModule 7-41

GetName 7-42, 10-67, 16-28, 16-64

GetObjectsByMaterial 8-229

GetObjectsInGroup 8-231

GetObjPath 7-43, 10-68, 16-29, 16-65

GetOutputVariableValue 7-44, 9-6

GetProjectDirectory 4-28

GetProperties 2-39, 6-34, 7-46

GetPropertyValue 2-40, 6-35, 7-47, 8-234, 10-68, 24-59

GetPropEvaluatedValue 8-234

GetPropNames 7-48

GetPropSIValue 8-237

GetRegistryInt 4-29, 4-62

GetRegistryString 4-30, 4-63

GetSelections 8-239

GetSolutionType 7-50

GetSolveInsideThreshold 7-50

GetSubGroupsInGroup 8-240

GetTempDirectory 4-33

GetVariables 2-41, 6-38, 7-51

GetVariableValue 2-42, 6-37, 7-51

GetVariationVariableValue 7-52

GetVersion 4-34

HasDataset 6-39

ImportIntoReport 10-87

IsFeaturEnabled 4-35

PasteDesign 7-57

PasteReports 10-91

PasteReportsWithLegacyNames 10-92

PasteTracesWithLegacyNames 10-93

Redo 7-58

RenameDesignInstance 7-59

RenameReport 10-93

RenameTraces 10-94

RunCTMAPI 7-60

RunToolkit 7-65

SARSetup 7-66

SavePlotSettingsAsDefault 10-95

SetActiveEditor 7-67

SetAllowMaterialOverride 7-68

SetLibraryDirectory 4-52

SetProjectDirectoryVBCommand> 4-53

SetPropertyValue 2-43, 6-54, 7-71, 24-68

SetSolutionType 7-72

SetSolveInsideThreshold 7-74

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

SetTempDirectory 4-56
SetVariableValue 2-44, 6-55, 7-75
ShowWindow 8-174
Solve 7-76
UpdateAllReports 10-98
ValidateDesign 7-77
Design Object Script Commands 7-1
Desktop Commands For Registry Values 4-60
Desktop object commands
AddMessage 4-6
ClearMessages 4-7
CloseProject 4-11
CloseProjectNoForce 4-12
Count 4-12
DeleteProject 4-13
DownloadJobResults 4-14
EnableAutosave 4-16
ExportOptionsFiles 4-17
FitSelected 7-33
GetActiveProject 4-17
GetAutosaveEnabled 4-18
GetBuildTimeDateString 4-18
GetChildNames 6-26, 7-34
GetChildNames Modeler 8-208
GetChildTypes 6-27, 7-36, 8-212
GetChildTypes Optimetrics 16-28, 16-64
GetCustomMenuSet 4-19
GetDesigns 6-29
GetDesktopConfiguration 4-22
GetDistributedAnalysisMachines 4-22
GetDistributedAnalysisMachinesForDesignType 4-23
GetProjectList 4-28
GetProjects 4-26, 4-29
GetPropNames 6-33, 16-31, 16-66
GetSchematicEnvironment 4-31
HideGeometry 7-53
KeepDesktopResponsive 4-36
LaunchJobMonitor 4-37
NewProject 4-37
OpenProject 4-38
PauseRecording 4-39
PauseScript 4-39
Print 4-40
QuitApplication 4-40
RefreshJobMonitor 4-41
ResetLogging 4-42
RestoreWindow 4-43
ResumeRecording 4-44
RunProgram 4-45
RunScript 4-45
SelectObjects 7-67
SelectScheduler 4-48
SetActiveProject 4-49

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

SetActiveProjectByPath 4-50
SetCustomMenuSet 4-50
SetDesktopConfiguration 4-51
SetSchematicEnvironment 4-55
ShowGeometry 7-76
Sleep 4-57
SubmitJob 4-58
TileWindows 4-59
Desktop Object Script Commands 4-1, 11-11
Desktop Scripting Conventions 1-56
DetachEdges 8-127
DetachFaces 8-128
DisableDiffPairs 23-12
DistributedAnalyzeSetup, Optimetrics module command 16-7, 16-40
DoesNamedExpressionExists 18-16, 19-10
DoesOutputVariableExist 9-3
DoesSupportTraceCharacteristics 10-39
Draw Menu commands 8-4
DumpAllReportsData 10-40
DuplicateAlongLine 8-86
DuplicateAroundAxis 8-88
DuplicateMirror 8-89

E

EcxmlExport_1 7-31
Edit 24-27, 24-48, 24-81, 24-105
Edit Menu Commands 8-83
Edit(BoundarySetup) 11-3
Edit3DComponent 8-54
Edit3DComponentDefinition 8-56
EditCartesianXMarker 10-40
EditCartesianYMarker 10-41
EditCoSimulationOptions 7-26
EditDataset 6-14, 24-40
EditDesignSettings 7-26
EditEntityList 8-129, 8-134
EditFaceCS 8-130
EditFieldsSummarySetting 20-1
EditGlobalMeshRegion 14-1
EditInfiniteArray 7-27
EditMarker 10-42
EditMaterial 6-16, 24-42
EditMeshRegion 14-9
EditNativeComponentDefinition 8-57
EditObjectCS 8-135
Editor object commands
SetPropertyValue 2-43, 6-54, 7-71, 24-68
EditOutputVariable 9-4
EditPolyline 8-60
EditRelativeCS 8-140
EditSetup 15-3, 16-8, 16-41
optimization command 16-79
parametric command 16-73
sensitivity command 16-88

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

statistical command 16-95
EditSources 17-3
EditSurfaceMeshSummaryData 18-16
EditUserDefinedSolution 22-3
EditWithComps 24-105
EMDesignOptions 7-28
EnableAutoSave 4-16
EnableDiffPairs 23-13
EnableSetup 16-19, 16-52
EnterComplex 18-18, 19-11
EnterComplexVector 18-19, 19-11
EnterCoord 18-20, 19-12
EnterEdge 18-20, 19-13
EnterLine 18-21, 19-13
EnterOutputVar 18-21, 19-14
EnterPoint 18-22, 19-14
EnterQty 18-23, 19-15
EnterScalar 18-23, 19-16
EnterScalarFunc 18-24, 19-16
EnterSurf 18-24, 19-17
EnterVector 18-25, 19-17
EnterVectorFunc 18-25, 19-18
EnterVol 18-26, 19-18
Example Scripts 26-1
Export 8-141, 24-54, 24-56, 24-57, 24-88, 24-114
ExportCableLibrary 8-268
ExportCitiFile 23-13
ExportConvergence 7-29
ExportDataset 6-23, 7-30, 24-54
ExportDOELocalSensitivity 16-61
ExportDOELocalSensitivityCurve 16-61
ExportDOEResponseCurve 16-59
ExportDOEResponseCurveSlices 16-59
ExportDOEResponseSurface 16-60
ExportDXConfigFile, Optimetrics module command 16-20, 16-52
ExportEyeMaskViolation 10-42
ExportFieldPlot 18-27
ExportFieldsSummary 20-2
ExportFullWaveSpice 23-15
ExportImageToFile 10-43
ExportLPVROM_1 7-30
ExportMarkerTable 18-27
ExportMaterial 6-24, 24-55
ExportMatlab 23-19
ExportModelMeshToFile 8-146, 10-51
ExportNetworkCSV 7-32
ExportNMFData 17-3
ExportOptimetricsProfile, Optimetrics module command 16-20, 16-53
ExportOptimetricsResults, Optimetrics module command 16-21, 16-54
ExportOptionsFiles 4-17
ExportOutputVariables 9-5
ExportParametricResults, Optimetrics module command 16-22, 16-55
ExportPlot3DToFile 10-51
ExportPlotImageToFile 18-30

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

ExportPlotImageWithViewToFile 18-31

ExportProfile 7-32

ExportReport 10-52

ExportReportDataToFile 10-53

ExportRespSurfaceMinMaxTable 16-23, 16-56

ExportRespSurfaceRefinePoints 16-24, 16-57

ExportRespSurfaceResponsePoints 16-25, 16-57

ExportRespSurfaceVerificationPoints 16-25, 16-58

ExportScript 24-57, 24-93

ExportSpreadsheet 23-22

ExportSurfaceMeshSummary 18-31

ExportTableToFile 10-54

ExportToFile 10-55, 18-33

ExportTouchstone 23-24

ExportTouchstone2 23-27

ExportTransientData 17-4

ExportUniformPointsToFile 10-57

Extract 23-29

F

FFTONReport 17-5

Field Calculator commands

DoesNamedExpressionExists 18-16, 19-10

Field Overlay module commands, GetFieldPlotNames 18-34

field overlay script commands 18-1

Field Overlays module commands

CalcOp 18-3, 19-3

CalcStack 18-4, 19-5

ChangeGeomSettings 18-6, 19-6

ClcMaterial 18-7, 19-7

ClearAllNamedExpr 18-9, 19-9

CopyNamedExprToStack 18-9, 19-9

DeleteFieldPlot 18-13

DeleteNamedExpr 18-15, 19-10

DeleteUneditablePlot 18-15

EditSurfaceMeshSummaryData 18-16

EnterComplex 18-18, 19-11

EnterComplexVector 18-19, 19-11

EnterCoord 18-20, 19-12

EnterEdge 18-20, 19-13

EnterLine 18-21, 19-13

EnterOutputVar 18-21, 19-14

EnterPoint 18-22, 19-14

EnterQty 18-23, 19-15

EnterScalar 18-23, 19-16

EnterScalarFunc 18-24, 19-16

EnterSurf 18-24, 19-17

EnterVector 18-25, 19-17

EnterVectorFunc 18-25, 19-18

EnteVol 18-26, 19-18

ExportFieldPlot 18-27

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

ExportOnGrid 18-28
ExportSurfaceMeshSummary 18-31
GetFieldFolderNames 18-34
GetFieldPlotQuantityName 18-35
GetMeshPlotNames 18-35
HidePolarPlot 18-37
HideRadiatedPlotOverlay 18-37
ModifyFieldPlot 18-39
ReassignFieldPlot 18-40
RenameFieldPlot 18-42
RenamePlotFolder 18-43
SaveFieldsPlots 18-43
SetFieldPlotSettings 18-45
SetPlotFolderSettings 18-46
UpdateAllFieldsPlots 18-50
UpdateQuantityFieldsPlots 18-50
Fields Calculator commands
AddNamedExpr 18-3, 19-2
CalcOp 18-3, 19-3
CalcStack 18-4, 19-5
CalculatorWrite 18-5, 19-5
ChangeGeomSettings 18-6, 19-6
ClcEval 18-7, 19-7
ClcMaterial 18-7, 19-7
ClcMaterialValue 18-8, 19-8
ClearAllNamedExpr 18-9, 19-9
CopyNamedExprToStack 18-9, 19-9
DeleteNamedExpr 18-15, 19-10
EnterComplex 18-18, 19-11
EnterComplexVector 18-19, 19-11
EnterCoord 18-20, 19-12
EnterEdge 18-20, 19-13
EnterLine 18-21, 19-13
EnterOutputVar 18-21, 19-14
EnterPoint 18-22, 19-14
EnterQty 18-23, 19-15
EnterScalar 18-23, 19-16
EnterScalarFunc 18-24, 19-16
EnterSurf 18-24, 19-17
EnterVector 18-25, 19-17
EnterVectorFunc 18-25, 19-18
EnterVol 18-26, 19-18
ExportOnGrid 19-19
ExportToFile 19-21
Fields Calculator Script Commands 19-1
Fields reporter module commands
AddMarker[Fields Reporter] 18-1
ClearAllMarkers[Fields Reporter] 18-8
DeleteMarker[Fields Reporter] 18-14
ExportMarkerTable 18-27
Fields Summary Script Commands 20-1
Fillet 8-147
FitSelected 7-33
FlattenGroup 8-148

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 8-205

GenerateConductivityFile 7-33

GenerateHistory 8-149

GenerateUserDefinedModel 8-205

GenerateVariationData Parametric, parametric command 16-26, 16-62, 16-74

GeometryCheckAndAutofix 8-206

Get3DComponentDefinitionNames 8-63

Get3DComponentInstanceNames 8-64

Get3DComponentMaterialNames 8-64

Get3DComponentMaterialProperties 8-65

Get3DComponentParameters 8-65

Get3DComponentPartNames 8-66

GetActiveDesign 6-24

GetActiveProject 4-17

GetAdaptiveSettings 17-6

GetAllCategories 10-58

GetAllQuantities 10-59

GetAllReportNames 10-60

GetAllSourceMagnitudes 17-6

GetAllSourceModes 17-7

GetAllSourcePhases 17-7

GetAllSources 17-8

GetAntennaParameters 17-8

GetArrayVariables 2-38, 6-25

GetAutoSaveEnabled 4-18

GetAvailableDisplayTypes 10-60

GetAvailableReportTypes 10-61

GetAvailableSolutions 10-61

GetBodyNamesByPosition 8-207

GetBoundaries 11-6

GetBoundariesOfType 11-6

GetBoundaryAssignment 11-7

GetBuildDateTimeString 4-18

GetChild Object (Report Setup), Report module command 10-62

GetChildNames 6-26, 7-34

GetChildNames (Optimetrics), Report module command 10-62

GetChildNames Modeler 8-208

GetChildObject Design 6-26, 7-35

GetChildObject Modeler 8-211

GetChildTypes 6-27, 7-36, 8-212

GetChildTypes Optimetrics 16-28, 16-64

GetChildTypes ReportSetup 10-63

GetCurvePropServerName 10-64

GetDataUnits 10-75

GetDefaultBaseName 11-7

GetDefinitionManager 6-28

GetDependentFiles 6-28

GetDesignID 7-36

GetDesigns 6-29

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

GetDesignType 7-37

GetDesignVariableNames 10-76

GetDesignVariableUnits 10-76

GetDesignVariableValue 10-77

GetDesignVariationKey 10-78

GetDisplayType 10-64

GetDistributedAnalysisMachines 4-22

GetDistributedAnalysisMachinesForDesignType 4-23

GetDocumentNames 21-5

GetDynLinkIntrinsicVariables 10-65

GetDynLinkQtyValueState 10-65

GetDynLinkTraces 10-66

GetDynLinkVariableValues 10-67

GetEdgeByPosition 8-214

GetEdgeIDFromNameForFirstOperation 8-215

GetEdgeIDsFromFace 8-216

GetEdgeIDsFromObject 8-216

GetEdgeLength 8-217

GetEdgePositionAtNormalizedParameter 7-37

GetEntityIDsContainedByNamedSelection 8-217

GetEntityListIDByName 8-218

GetExeDir 4-23

GetExtendedDefinitionObect 8-218

GetFaceArea 8-219

GetFaceByPosition 8-220

GetFaceCenter 8-219

GetFaceIDFromNameForFirstOperation 8-221

GetFaceIDs 8-222

GetFaceIDsOfSheet 8-222

GetFaceMeshCentroidAndArea 17-9

GetFaceMeshIDAndArea 17-9

GetFieldFolderNames 18-34

GetFieldPlotNames, Field Overlay module command 18-34

GetFieldPlotQuantityName 18-35

GetFieldType 17-10

GetFrequencies 23-30

GetFrequencyCount 23-30

GetGeometryIdsForAllNetLayerCombinations 7-39

GetGeometryIdsForNetLayerCombinations 7-38

GetGeometryModelerMode 8-223

GetGroupSubmodelNames 8-223

GetImagDataValues 10-79

GetIncludePortPostProcessing 17-10

GetLibraryDirectory 4-24

GetMatchedObjectName 8-224

GetMeshPlotNames 18-35

GetModelBoundingBox 8-225

GetModelUnits 8-225

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

GetModule 7-41

GetMultipactionBreakdown 17-11

GetName 6-31, 7-42, 10-67, 16-28, 16-64, 23-31

GetNames 24-71

GetNetworkDataSolution 17-12

GetNetworkDataSolutionDefinition 17-12

GetNetworkPostprocSetup 17-13

GetNumBoundaries 11-8

GetNumBoundariesOfType 11-8

GetNumObjects 8-226

GetObjectIDByName 8-226

GetObjectName 8-227

GetObjectNameByEdgeID 8-227

GetObjectNameByFaceID 8-228

GetObjectNameByID 8-228

GetObjectNameByVertexID 8-229

GetObjectsByMaterial 8-229

GetObjectShapeType 8-230

GetObjectsIngroup 8-231

GetObjectVolume 8-231

GetObjPath 6-32, 7-43, 8-232, 10-68, 16-29, 16-65

GetOptimetricsResults, Optimetrics module command 16-30

GetOutputVariableValue 7-44, 9-6

GetPartsForUserDefinedModel 8-233

GetPath 6-32

GetPerQuantityPrimarySweepValues 10-80

GetPersonalLibDirectory 4-27

GetPoints 8-233

GetPortCount 23-32

GetPortNumber 23-33

GetPostProcSettings 23-34

GetProcessID 4-27

GetProjectDirectory 4-28

GetProjectList 4-28

GetProjectMaterialNames 24-58

GetProjects 4-26, 4-29

GetProperties 2-39, 6-34, 7-46, 24-72

GetPropertyValue 2-40, 6-35, 7-47, 8-234, 10-68, 24-59

GetPropEvaluatedValue 8-234

GetPropNames 6-33, 7-48, 10-70, 16-31, 16-66

GetPropNames Modeler 8-236

GetPropSIValue 8-237

GetPropValue Modeler 8-238

GetPropValue Optimetrics 16-31, 16-67

GetPropValue Project 6-34, 7-49

GetPropValue Reporter 10-70

GetQtyExpressionsForSourceTrace 10-71

GetRealDataValues 10-80

GetRegistryInt 4-29, 4-62

GetRegistryString 4-30, 4-63

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

GetRelativeCoordinateSystems 8-239

GetReportSummaryForRegressionTesting 10-72

GetReportTraceNames 10-71

GetSelections 7-50, 8-239

GetSetupCount 15-15

GetSetupCount, Analysis module command 15-15

GetSetupNames (Optimetrics), Optimetrics module command 16-26, 16-27, 16-32, 16-63, 16-63, 16-67

GetSetupNamesByType (Optimetrics), Optimetrics module command 16-32, 16-68

GetSetups 15-15

Analysis module command 15-15

GetSolutionContexts 10-72

GetSolutionDataPerVariation 10-73

GetSolutionType 7-50

GetSolutionVariation 23-34

GetSolveInsideThreshold 7-50

GetSourceContexts 17-13

GetSubGroupsInGroup 8-240

GetSweepNames 10-81

GetSweepUnits 10-82

GetSweepValues 10-83

GetSysLibDirectory 4-33

GetTempDirectory 4-33

GetTerminalExcitationType 17-14

GetTool 4-65

GetTopDesignList 6-37

GetTopEntryValue 18-36, 19-24

GetTransientSolveTimes 17-14

GetUserDefinedSolutionNames 22-4

GetUserDefinedSolutionProperties 22-4

GetUserLibDirectory 4-34

GetUserPosition 8-240

GetVariables 2-41, 6-38, 7-51

GetVariableValue 2-42, 6-37, 7-51

GetVariation 23-35

GetVariationVariableValue 7-52

GetVersion 4-34

GetVertexIDFromNameForFirstOperation 8-241

GetVertexIDsFromEdge 8-242

GetVertexIDsFromFace 8-242

GetVertexIDsFromObject 8-243

GetVertexPosition 8-243

GetWireBodyNames 8-244

GroupPlotCurvesByGroupingStrategy 10-86

H

HasDataset 6-39

HasSameData 23-36

HealObject 8-150

HideGeometry 7-53

HidePolarPlot 18-37

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

HideRadiatedPlotOverlay 18-37

hierarchy of variables in HFSS 1-49

I

Icepak boundary condition commands

AssignAdiabaticPlateBoundary 11-18

AssignBlockBoundary 11-12

AssignBlowerBoundary_1 11-29

AssignConductingPlateBoundary 11-19

AssignEMLoss 11-27

AssignFaceMonitor 13-1

AssignGrilleBoundary 11-14

AssignHeatExchangerBoundary 11-30

AssignInitialTemperature 11-32

AssignMeshOperation 14-5, 14-6

AssignMeshRegion 14-3, 14-7

AssignNetworkBoundary 11-15

AssignOpeningBoundary 11-16

AssignPointMonitor 13-2

AssignRecircBoundary 11-17

AssignResistanceBoundary 11-21

AssignSolarLoading 11-31

AssignSourceBoundary 11-24

AssignStationaryWallBoundary 11-25

AssignSymmetryWallBoundary 11-27

CreatePcbPowermap 7-9

EcxmlExport 7-31

EditGlobalMeshRegion 14-1

ExportDOELocalSensitivity 16-61

ExportDOELocalSensitivityCurve 16-61

ExportDOEResponseCurve 16-59

ExportDOEResponseCurveSlices 16-59

ExportDOEResponseSurface 16-60

ExportLPVROM 7-30

ExportNetworkCSV 7-32

ImportECXML 4-80

ImportECXML_1 4-81

ImportIDX 4-76

ImportNetworkCSV 7-55

InsertNativeComponent 12-1, 12-5, 12-8, 12-15, 12-20

InsertSetup 15-16, 15-24

Icepak component commands

GenerateConductivityFile 7-33

Import (3D Modeler command) 8-152

ImportANF 4-66, 4-67

ImportAutoCAD 4-68

ImportAWRMicrowaveOffice 4-68

ImportCableLibrary 8-269

ImportDataset 6-39, 7-53, 24-60

ImportDXF 8-155

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

ImportECXML_1 4-80
ImportEDB 4-69
ImportExport Tool 4-65
ImportExtracta 4-70
ImportFromClipboard 8-157
ImportGDSII 4-71, 4-72, 8-158
ImportIDF 4-72, 4-75
ImportIDX 4-76
ImportIntoReport 10-87
ImportIPC 4-78
ImportJEDEC 4-81
ImportNetworkCSV 7-55
ImportODB 4-79
ImportOutputVariables 9-7
ImportReportDataIntoReport 10-87
ImportSetup, Optimetrics module command 16-33, 16-69
ImportXFL 4-80
Imprint 8-160
ImprintProjection 8-161
Insert3DComponent 8-66
InsertComponent 8-68
InsertDesign 6-41, 7-57
InsertDesignWithWorkflow 6-42
InsertNativeComponent 8-69, 12-1, 12-5, 12-8, 12-15, 12-20
InsertSetup 15-16, 15-24
optimization command 16-82
parametric command 16-75
sensitivity command 16-92
statistical command 16-95
Intersect 8-163
IronPython 1-2
IsDataComplex 10-84
IsFeatureEnabled 4-35
IsPerQuantityPrimarySweep 10-84
IsUsed 24-89

K

KeepDesktopResponsive 4-36

L

LaunchJobMonitor 4-36
Layout Scripting 1-58
LoadNamedExpressions 18-38, 19-24
LoadSolution 23-38

M

material commands
AddDefinitionFromBlock 8-190, 24-7
AddDefinitionFromLibFile 8-192
AddMaterial 6-6, 24-9
EditMaterial 6-16, 24-42
ExportMaterial 6-24, 24-55
GetExtendedDefinitionObject 8-218
RemoveMaterial 6-45, 24-64
UpdateDefFromBlock 24-69
Mesh Operations module commands
EditMeshRegion 14-9

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

Mesh Operations Module Script Commands 14-1

Mirror 8-92

Model Manager Script Commands 24-74

Modeler Menu Commands 8-98

ModifyFieldPlot 18-39

module script commands

boundary conditions 11-1

field overlay 18-1

modules in HFSS scripting 1-49

Move 8-93

MoveCSToEnd 8-164

MoveEnityToGroup 8-164

MoveFaces 8-165

MovePlotCurvestoGroup 10-88

MovePlotCurvestoNewGroup 10-89

N

Named Arguments 1-56

NdExplorer

Script Commands 23-1

NewProject 4-37

O

oAnsoftApp object 1-49

object-oriented scripting 2-1

oDesign object 1-49

oDesktop object 1-49

oEditor object 1-49

OffsetFaces 8-94

oModule object 1-49

Open 23-38

OpenProject 4-38

OpenWindowForAllReports 10-89

OpenWindowForReports 10-90

oProject object 1-49

Optimetrics module commands

DeleteSetups 16-7, 16-39

DistributeAnalyzeSetup 16-7, 16-40

EnableSetup 16-19, 16-52

ExportDXConfigFile 16-20, 16-53

ExportOptimetricsProfile 16-20, 16-53

ExportOptimetricsResults 16-21, 16-54

ExportParametricResults 16-22, 16-55

ExportRespSurfaceMinMaxTable 16-23, 16-56

ExportRespSurfaceRefinePoints 16-24, 16-57

ExportRespSurfaceResponsePoints 16-25, 16-57

ExportRespSurfaceVerificationPoints 16-25, 16-58

GetChildNames 16-26, 16-27, 16-63, 16-63

GetOptimetricResult 16-29, 16-65

GetOptimetricsResults 16-30

GetSetupNames 16-32, 16-67

GetSetupNamesByType 16-33,

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

16-68

ImportSetup 16-33, 16-69

RenameSetup 16-35, 16-70

SolveSetup 15-2, 16-6, 16-36, 16-37, 16-39, 16-71, 16-72

Optimetrics Module Script Commands 16-1

optimization commands

EditSetup 16-79

InsertSetup 16-83

Optimization Script Commands 16-79

Other oEditor Commands 8-187

output variable commands

CreateOutputVariable 9-1

DeleteOutputVariable 7-24, 7-25, 9-2

DoesOutputVariableExist 9-3

EditOutputVariable 9-4

GetOutputVariableValue 7-44, 9-6

Output variable commands

ExportOutputVariables 9-5

ImportOutputVariables 9-7

Output Variable Script Commands 9-1

P

PageSetup 8-244

parametric commands

EditSetup 16-73

GenerateVariationData Parametric 16-26, 16-62, 16-74

InsertSetup 16-75

Parametric Script Commands 16-73

Paste 6-43, 8-266, 8-267

Paste (Model Editor) 8-95

Paste (Project Object) 6-43

PasteDesign 7-57

PastePlotSettings 10-90

PasteReports 10-91

PasteReportsWithLegacyNames 10-92

PasteSetup

Analysis module command 15-33

Optimetrics module command 16-34, 16-69

PasteTraces 10-92

PasteTracesWithLegacyNames 10-93

PauseRecording 4-39

PauseScript 4-39

Print 4-40

Project object commands

AddDataset 6-4, 7-4, 24-5

AddMaterial 6-6, 24-9

Close 6-10

CopyDesign 6-11

CutDesign 6-12

DeleteDataset 6-12, 7-23, 24-27

DeleteDesign 6-13

EditDataset 6-14, 6-23, 6-39, 7-30, 7-53, 24-40, 24-54, 24-60

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

EditMaterial 6-16, 24-42
ExportMaterial 6-24, 24-55
GetActiveDesign 6-24
GetArrayVariables 2-38, 6-25
GetChildObject Design 6-26, 7-35
GetChildObject Modeler 8-211
GetDependentFiles 6-28
GetDesignID 7-36
GetDesignType 7-37
GetName 6-31
GetObjPath 6-32
GetPath 6-32
GetProjectMaterialNames 24-58
GetProperties 2-39, 6-34, 7-46
GetPropertyValue 2-40, 6-35, 7-47, 8-234, 10-68, 24-59
GetPropEvaluatedValue 8-234
GetPropNames Modeler 8-236
GetPropSIValue 8-237
GetPropvalue Modeler 8-238
GetPropvalue Optimetrics 16-31, 16-67
GetPropvalue Project 6-34, 7-49
GetPropvalue Reporter 10-70
GetSolutionVariation 23-34
GetTopDesignList 6-37
GetTopEntryValue 18-36, 19-24
GetVariables 2-41, 6-38, 7-51
GetVariableValue 2-42, 6-37, 7-51
InsertDesign 6-41, 7-57
LoadSolution 23-38
Paste 6-43, 6-43
Redo 6-44
RemoveAllUnusedDefinitions 6-44
RemoveMaterial 6-45, 24-64
RemoveUnusedDefinitions 6-46, 24-67
Rename 6-47
RestoreProjectArchive 4-42
Save 6-47
SaveAs 6-48
SaveAsStandAloneProject 6-50
SaveProjectArchive 6-51
SetActiveDesign 6-52
SetPropertyValue 2-43, 6-54, 7-71, 24-68
SetPropValue Design 7-70
SetPropValue Optimetrics 16-35, 16-71
SetPropValue Project 6-53, 10-97
SetVariableValue 2-44, 6-55, 7-75
SimulateAll 6-9, 6-56
Undo 6-56
UpdateDefinitions 6-57
Project Object Script Commands 6-1
ProjectSheet 8-167
property commands
GetArrayVariables 2-38, 6-25

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

GetProjectMaterialNames 24-58
GetProperties 2-39, 6-34, 7-46
GetPropertyValue 2-40, 6-35, 7-47, 8-234, 10-68, 24-59
GetPropEvaluatedValue 8-234
GetPropSIValue 8-237
GetTopEntryValue 18-36, 19-24
GetVariables 2-41, 6-38, 7-51
GetVariableValue 2-42, 6-37, 7-51
SetPropertyValue 2-43, 6-54, 7-71, 24-68
SetVariableValue 2-44, 6-55, 7-75
Property Script Commands 2-25
Conventions uSed in thie Chapter 2-36
Object Script Property Function Summary 2-27
PurgeHistory 8-168

Q

QuitApplication 4-40

R

ReassignBoundaries 11-2, 11-8
ReassignBoundary 11-8
ReassignFieldPlot 18-40
Redo
design-level command 7-58
project-level command 6-44
RefreshJobMonitor 4-41
ReleaseData 10-85
Remove 24-62, 24-63, 24-65, 24-66, 24-89, 24-116
RemoveAllUnusedDefinitions 6-44
RemoveAssignmentFromBoundary 11-9
RemoveBadEdges 8-246
RemoveBadFaces 8-246
RemoveBadVertices 8-247
RemoveCables 8-269
RemoveMaterial 6-45, 24-64
RemoveUnusedDefinitions 6-46, 24-67
Rename 6-47, 23-39
RenameBoundary 11-10
RenameDesignInstance 7-59
RenameFieldPlot 18-42
RenamePart 8-247
RenamePlotFolder 18-43
RenameReport 10-93
RenameSetup 15-33
Analysis module command 15-33
Optimetrics module command 16-35, 16-70
RenameSource [Interface Source] 7-59
RenameTraces 10-94
Renormalize 23-40
Reorder 23-41, 23-42
Replacewith3DComponent 8-168
Report module commands
GetChildNames 10-62

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

Reporter editor commands

AddAllEyeMeasurements 10-4

AddCartesianLimitLine 10-4

AddCartesianLimitLineFromCurve 10-5

AddCartesianLimitLineFromEquation 10-7

AddCartesianXMarker 10-8, 10-9

AddCartesianYMarkerToStack 10-9

AddDeltaMarker 10-10

AddMarker 10-11

AddMarkerToPlot 18-2

AddNote 10-12

AddTraceCharacteristics 10-13

AddTraces 10-14

ApplyReportTemplate 10-16

ClearAllMarkers 10-17

ClearAllTraceCharacteristics 10-17

CloneReportsFromDatasetSolution 10-18

CopyPlotSettings 10-19

CopyReportDefinition 10-19

CopyReportsData 10-20

CopyTraceDefinitions 10-20

CopyTracesData 10-21

CreateReport 7-10, 10-22

CreateReportFromTemplate 10-34

CreateReportOfAllQuantities 10-35

DeleteAllReports 10-37

DeleteReport 10-37

DeleteTraceCharacteristics 10-38

DeleteTraces 10-38

DoesSupportTraceCharacteristics 10-39

DumpAllReportsData 10-40

EditCartesianXMarker 10-40

EditCartesianYMarker 10-41

EditMarker 10-42

ExportEyeMaskViolation 10-42

ExportImageToFile 10-43

ExportPlot3DToFile 10-51

ExportPlotImageToFile 18-30, 18-31

ExportReport 10-52

ExportReportDataToFile 10-53

ExportTableToFile 10-54

ExportToFile 10-56, 10-86

ExportUniformPointsToFile 10-57

FFTOnReport 17-5

GetAllCategories 10-58

GetAllQuantities 10-59

GetAllReportNames 10-60

GetAvailableDisplayTypes 10-60

GetAvailableReportTypes 10-61

GetAvailableSolutionss 10-61

GetChildTypes ReportSetup 10-63

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

GetCurvePropServerName 10-64
GetDisplayType 10-64
GetDynLinkIntrinsicVariables 10-65
GetDynLinkQtyValueState 10-65
GetDynLinkTraces 10-66
GetDynLinkVariableValues 10-67
GetPropNames 10-70
GetQtyExpressionsForSourceTrace 10-71
GetReportSummaryForRegressionTesting 10-72
GetReportTraceNames 10-71
GetSolutionContexts 10-72, 10-73
GroupPlotCurvesByGroupingStrategy 10-86
ImportReportDataIntoReport 10-87
MovePlotCurvestoGroup 10-88
MovePlotCurvestoNewGroup 10-89
OpenWindowForAllReports 10-89
OpenWindowForReports 10-90
PastePlotSettings 10-90
PasteReports 10-91
PasteReportsWithLegacyNames 10-92
PasteTraces 10-92
PasteTracesWithLegacyNames 10-93
RenameReport 10-93
RenameTraces 10-94
ResetPlotSettings 10-95
SavePlotSettingsAsDefault 10-95
SetLinkOutputTraces 10-96
UnGroupPlotCurvesInGroup 10-97
UpdateAllReports 10-98
UpdateReports 10-98
UpdateTraces 10-99
UpdateTracesContextAndSweeps 10-101
Reporter Editor Script Commands 10-1
Reporter module commands
GetChildObject 10-62
ReprioritizeBoundaries 11-10
Reset 23-43
ResetLogging 4-42
ResetPlotSettings 10-95
RestoreWindow 4-43
ResumeRecording 4-44
RevertAllToInitial 15-33
RevertSetupToInitial 15-34
RevertSetupToInitialCondition 15-35
Rotate 8-96
RunCTMAPI 7-60
Running Instance Manager Script Commands 5-1

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

RunProgram 4-45
RunScript 4-45
RunScriptWithArguments 4-47
RunToolkit 7-65

S

sample scripts
  variable helix 26-6
  waveguide 26-9
SARSetup 7-66
Save 6-47
SaveAs 6-48
SaveAsStandAloneProject 6-50
SaveFieldsPlots 18-43
SaveNamedExpressions 18-44, 19-25
SavePlotSettingsAsDefault 10-95
Scale 8-97
Script and Library Scripts 24-91
script commands
  boundary/excitation 11-1
  field overlay 18-1
Script Commands for Creating and Modifying Mesh Regions 14-1
scripting
  object-oriented 2-1
scripts
  CPython 1-44
  IronPython 1-2
  overview 1-1
  pausing 1-53
  recording 1-53
  running 1-52
  stopping 1-52
Scripts and Locked Layers 1-59
Section 8-170
SelectObjects 7-67
SelectScheduler 4-13, 4-48
sensitivity commands
  EditSetup 16-88
  InsertSetup 16-92
Sensitivity Script Commands 16-88
SeparateBody 8-172
SetActiveDefinitionEditor 6-52
SetActiveDesign 6-52
SetActiveEditor 7-67
SetActiveProject 4-49
SetActiveProjectByPath 4-50
SetAllowMaterialOverride 7-68
SetAllPortImpedance 23-44
SetBackgroundMaterial 7-69
SetDefaultBaseName 11-11
SetFieldPlotSettings 18-45
SetLengthSettings 7-69
SetLibraryDirectory 4-52
SetLinkOutputTraces 10-96
SetModelUnits 8-172
SetPlotFolderSettings 18-46
SetPortDeembedDistance 23-45

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

SetPortImpedance 23-46

SetPostProcSettings 23-47

SetProjectDirectory 4-53

SetPropertyValue 2-43, 6-54, 7-71, 24-68

SetPropValue Design 7-70

SetPropValue Modeler 8-248

SetPropValue Optimetrics 16-35, 16-71

SetPropValue Project 6-53, 10-97

SetSolutionType 7-72

SetSolveInsideThreshold 7-74

SetSourceContexts 7-74, 17-15

SetTempDirectory 4-56

Setting Numerical Values 1-58

SetTopDownViewDirectionForActiveView 8-253

SetTopDownViewDirectionForAllViews 8-254

SetVariableValue 2-44, 6-55, 7-75

SetWCS 8-173

SGetAppDesktop 3-1

ShowGeometry 7-76

ShowIWindow 8-174

Simplify 8-175

SimulateAll 6-9, 6-56

SimValueContext 9-7

Sleep 4-57

Smooth 23-48

Solution module commands

GetSourceContexts 17-13

SetSourceContexts 7-74, 17-15

Solutions module commands

ConstructVariationString 7-8

DeleteSolutionVariation 7-24, 17-2

EditSources 17-3

ExportNMF 17-3

ExportTransientData 17-4

GetAdaptiveSettings 17-6

GetAllSourceMagnitudes 17-6

GetAllSourceModes 17-7

GetAllSourcePhases 17-7

GetAllSources 17-8

GetAntennaParameters 17-8

GetFieldType 17-10

GetIncludePortPostProcessing 17-10

GetMultipactionBreakdown 17-11

GetNetworkDataSolution 17-12

GetNetworkDataSolutionDefinition 17-12

GetNetworkPostprocSetup 17-13

GetTerminalExcitationType 17-14

GetTransientSolveTimes 17-14

TDROnReport 17-16

Solutions Module Script Commands 17-1

SolveAllSetup, Optimetrics module command 16-36, 16-71

SolveSetup, Optimetrics module command 16-37, 16-72

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

Split 8-177

statistical commands

EditSetup 16-95

InsertSetup 16-95

Statistical Script Commands 16-95

Stitch 8-179

Stretch 23-49

Structures 1-30

SubmitJob 4-58

Subtract 8-180

SweepAlongPath 8-75

SweepAlongVector 8-76

SweepAroundAxis 8-78

SweepFacesAlongNormal 8-80, 8-80, 8-181, 8-181

SweepFacesAlongNormalWithAttributes 8-81

Symbol Manager Script Commands 24-96

T

TDROnReport 17-16

Terminate 23-50

Thermal Monitor 13-1

ThickenSheet 8-182

U

UDM Structures 1-30

UncoverFaces 8-183

Undo

project-level command 6-56

Ungroup 8-186

UnGroupPlotCurvesInGroup 10-97

Unite 8-184

UpdateAllFieldsPlots 18-50

UpdateAllReports 10-98

UpdateCableHarness 8-270

UpdateComponentDefinition 8-83

UpdateDefFromBlock 24-69

UpdateDefinitions, project-level command 6-57

UpdatePriorityList 8-254

UpdateQuantityFieldsPlots 18-50

UpdateReports 10-98

UpdateTraces 10-99

UpdateTracesContextAndSweeps 10-101

UpgradeVersion 8-255

User defined documents module commands

GetDocumentNames 21-5

User defined solution module commands

CreateUserDefinedSolution 22-1

DeleteUserDefinedSolutions 22-2

EditUserDefinedSolution 22-3

GetUserDefinedSolutionNames 22-4

GetUserDefinedSolutionProperties 22-4

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 Solutions Commands 22-1

V

Validate3DComponent 8-256

ValidateDesign 7-77

variables

hierarchy in HFSS 1-49

used in HFSS scripts 1-49

W

WrapSheet 8-186

WriteHistoryTreeLayoutForTest 8-257

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