

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

Circuit 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 with Python 1-5

Scripting using Embedded VBScript or JavaScript 1-5

Standalone IronPython 1-7

Running Standalone IronPython 1-7

Using a Recorded Script 1-7

Creating an External Script 1-7

Example Script 1-8

IronPython Samples 1-9

Change property 1-10

Create a Cone using IronPython 1-11

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

Advantages Compared to C++ 1-15

Changes compared to C 1-15

Structures 1-15

Return Values for UDM and UDP Functions 1-15

Constants 1-15

Methods 1-15

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

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

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

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

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

Electronics Desktop Scripting Conventions ... 1-57
Named Arguments ... 1-57
IronPython Example ... 1-58
Setting Numerical Values ... 1-59
Layout Scripts and the Active Layer ... 1-59
Scripts and Locked Layers ... 1-60
2 - Object-Oriented Scripting ... 2-1
Object-Oriented Scripting Logic/Syntax ... 2-1
Basic Functions Overview ... 2-1
Body Properties and Modification ... 2-8
Retrieving Variables ... 2-8
Retrieving Datasets and Values ... 2-9
GetSolutionData API ... 2-10
Summary ... 2-11
Additional Details Specific to AEDT Solvers ... 2-14
Additional Details on Boundaries/Excitations ... 2-15
3D Component Encapsulation ... 2-16
Additional Details on Solve Setup ... 2-17
Object-Oriented Scripting for Editors ... 2-18
Schematic Editor ... 2-18
Object-Oriented Scripting for Materials ... 2-18
Example Change to Material Property Type: ... 2-22
Example Change to Vector Component Value: ... 2-23
Example Change to Choice Property Value: ... 2-24
Property Script Commands ... 2-25
Object Script Property Function Summary ... 2-27
Object Path ... 2-27
Property Object ... 2-27
Project Object ... 2-29
Design Object ... 2-30

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

3D Modeler Object 2-30
Variable Object 2-31
Optimetrics Module Object: 2-32
Optimetrics Setup Object 2-33
ReportSetup(Results) Module Object: 2-33
ReportSetup(Results) Module Child Objects: 2-34
Radiation Module Child Objects: 2-35
Conventions Used in this Chapter 2-36
Callback Scripting Using PropHost Object 2-38
ChangeProperty (Schematic Editor and Layout Editor) 2-41
Property Array Structure 2-43
PropHost Functions 2-48
AddMenuProp 2-48
AddMenuProp2 2-49
AddProp 2-51
AddProp2 2-54
ExecuteScript 2-56
GetApplication 2-56
GetCallback 2-57
GetChangedProperty 2-58
GetDescription 2-58
GetDesign 2-59
GetEditor 2-59
GetEvaluatedText 2-60
GetFileName 2-60
GetHidden 2-61
GetProgress 2-62
GetPropHost 2-62
GetPropServers 2-62
GetPropTabType 2-63

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

GetReadOnly ... 2-63
GetRunStatus ... 2-64
GetTabTypeName ... 2-64
GetText ... 2-65
GetValue ... 2-66
IsValueConstant ... 2-66
PropertyExists ... 2-67
RemoveProp ... 2-67
SetCallback ... 2-68
SetDescription ... 2-69
SetHidden ... 2-69
SetReadOnly ... 2-70
SetText ... 2-70
SetValue ... 2-71
GetArrayVariables ... 2-72
GetProperties ... 2-72
GetPropertyValue ... 2-73
GetVariables ... 2-75
GetVariableValue ... 2-75
SetPropertyValue ... 2-76
SetVariableValue ... 2-77
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

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

DeleteProject ... 4-13
DownloadJobResults ... 4-14
DeleteRegistryEntry ... 4-15
DoesRegistryValueExist ... 4-15
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

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

GetTempDirectory 4-33
GetUserLibDirectory 4-34
GetVersion 4-34
IsFeatureEnabled 4-35
KeepDesktopResponsive 4-36
LaunchJobMonitor 4-37
NewProject 4-37
OpenAndConvertProject 4-38
OpenProject 4-39
OpenProjectWithConversion 4-39
PageSetup (Layout Editor) 4-40
PauseRecording 4-40
PauseScript 4-41
Print 4-41
QuitApplication 4-42
RefreshJobMonitor 4-42
ResetLogging 4-43
RestoreProjectArchive 4-44
RestoreWindow 4-45
ResumeRecording 4-45
RunACTWizardScript 4-46
RunProgram 4-46
RunScript 4-47
RunScriptWithArguments 4-48
SelectScheduler 4-49
SetActiveProject 4-51
SetActiveProjectByPath 4-51
SetCustomMenuSet 4-52
SetDesktopConfiguration 4-53
SetLibraryDirectory 4-54

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

SetProjectDirectory ... 4-54
SetRegistryFromFile ... 4-55
SetRegistryInt ... 4-55
SetRegistryString ... 4-56
SetSchematicEnvironment ... 4-57
SetTempDirectory ... 4-57
ShowDockingWindow ... 4-58
Sleep ... 4-59
SubmitJob ... 4-59
TileWindows ... 4-60
Desktop Commands For Registry Values ... 4-61
DeleteRegistryEntry ... 4-62
DoesRegistryValueExist ... 4-63
GetRegistryInt ... 4-63
GetRegistryString ... 4-64
SetRegistryFromFile ... 4-65
SetRegistryInt ... 4-65
SetRegistryString ... 4-66
ImportExport Tool Commands ... 4-67
ImportANF ... 4-68
ImportANFv2 ... 4-68
ImportAutoCAD ... 4-69
ImportAWRMicrowaveOffice ... 4-70
ImportEDB ... 4-71
ImportExtracta ... 4-71
ImportGDSII ... 4-72
ImportGerber ... 4-73
ImportIDF ... 4-74
ImportIDFandMerge ... 4-77
ImportIDX ... 4-77

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

ImportIPC ... 4-80
ImportODB ... 4-80
ImportXFL ... 4-81
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 ... 6-5
AnalyzeAll [project] ... 6-8
ChangeProperty ... 6-9
ClearMessages ... 6-13
CloneMaterial ... 6-13
Close ... 6-14
CopyDesign ... 6-15
CutDesign ... 6-15
DeleteDataset ... 6-16
DeleteDesign ... 6-17
DeleteToolObject ... 6-17
EditDataset ... 6-18
EditMaterial ... 6-19
ExportDataset ... 6-26
ExportMaterial ... 6-27
GetActiveDesign ... 6-27
GetArrayVariables ... 6-28
GetChildNames [Project] ... 6-29
GetChildObject [Project] ... 6-29
GetChildTypes [Project] ... 6-30
GetDefinitionManager ... 6-31

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

GetDependentFiles 6-31
GetDesign 6-32
GetDesigns 6-32
GetEDBHandle 6-33
GetLegacyName 6-34
GetName [Project] 6-34
GetObjPath [Project] 6-35
GetPath 6-35
GetPropEvaluatedValue 6-36
GetPropNames [Project] 6-37
GetPropSIValue 6-38
GetPropValue [Project] 6-39
GetProperties 6-39
GetPropertyValue 6-40
GetTopDesignList 6-42
GetVariableValue 6-42
GetVariables 6-43
ImportDataset 6-44
Note About File Types: 6-44
InsertDesign 6-45
InsertDesignWithWorkflow 6-46
InsertToolObject 6-47
Paste (Project Object) 6-48
Redo [Project Level] 6-48
RemoveAllUnusedDefinitions 6-49
RemoveMaterial 6-49
RemoveUnusedDefinitions 6-50
Rename 6-51
RunToolkit 6-52
Save 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.

SaveAs ... 6-53
SaveAsStandAloneProject ... 6-55
SaveProjectArchive ... 6-56
SetActiveDefinitionEditor ... 6-57
SetActiveDesign ... 6-57
SetPropValue [Project] ... 6-58
SetPropertyValue ... 6-59
SetVariableValue ... 6-60
SimulateAll ... 6-61
Undo [Project] ... 6-61
UpdateDefinitions ... 6-62
7 - Design Object Script Commands ... 7-1
AddDataset ... 7-5
AddDynamicLink ... 7-6
AddModelingProperties ... 7-8
AddMaxwellLink ... 7-8
AddQ3DRLGCLink ... 7-9
AnalyzeAll [design] ... 7-10
AnalyzeAllNominal ... 7-10
ChangePortProperty [Nexxim] ... 7-11
ChangeProperty ... 7-12
ChangeSourceProperty [Nexxim] ... 7-16
CopyEyeItemAsCommand ... 7-17
CopyItemCommand ... 7-18
CreateReport ... 7-18
DeleteDataset ... 7-31
DeleteDesignInstance ... 7-32
DeleteOutputVariable ... 7-32
DeletePort [Nexxim] ... 7-33
DeleteSolutionVariation ... 7-33

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

DeleteSource [Nexxim] 7-34
EditDataset 7-35
EditImportData 7-36
EditNotes 7-37
EditOutputVariable 7-37
ExportDataset 7-39
ExportForSpice 7-39
ExportNMFData 7-40
ExportNetlist [Nexxim] 7-40
ExportReport 7-41
GetAllPorts 7-42
GetChildNames [Design] 7-42
GetChildObject [Design] 7-43
GetChildTypes [Design] 7-44
GetDesignID 7-44
GetDesignType 7-45
GetEditor 7-45
GetLogFile 7-46
GetModule 7-46
GetName 7-47
GetNoteText 7-48
GetObjPath [Design] 7-49
GetOutputVariableValue 7-49
GetOutputVariables 7-50
GetPropEvaluatedValue 7-51
GetPropHost 7-52
GetPropNames [Design] 7-52
GetPropSIValue 7-53
GetPropValue [Design] 7-54
GetProperties 7-54

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

GetPropertyValue 7-55
GetResultsDirectory [Nexxim] 7-57
GetSourceData (Layout Editor) 7-58
GetVariableValue 7-58
GetVariables 7-59
GetVariationVariableValue 7-59
ImportData [Nexxim] 7-60
ImportDataFilePath [Nexxim] 7-63
ImportDataset 7-64
Note About File Types: 7-65
InsertDesign 7-66
PasteDesign (Design Object) 7-67
PasteItemCommand 7-68
Redo [Design] 7-69
RemoveImportData (Layout Editor) 7-69
RenameDesignInstance 7-70
RenameImportData [Nexxim] 7-70
RenameSource [Nexxim] 7-71
RunToolkit 7-72
SetActiveEditor 7-72
SetPropValue [Design] 7-73
SetPropertyValue 7-74
SetVariableValue 7-75
SimulateLink 7-76
Undo [Design] 7-76
UseCircuitSParameterDefinition 7-77
8 - Output Variable Script Commands 8-1
CreateOutputVariable 8-1
DeleteOutputVariable 8-3
DoesOutputVariableExist 8-3

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

EditOutputVariable ... 8-4
ExportOutputVariables ... 8-5
GetOutputVariables ... 8-6
GetOutputVariableValue ... 8-7
ImportOutputVariables ... 8-8
SimValueContext ... 8-9
9 - Reporter Editor Script Commands ... 9-1
AddAllEyeMeasurements ... 9-4
AddCartesianLimitLine ... 9-4
AddCartesianLimitLineFromCurve ... 9-5
AddCartesianLimitLineFromEquation ... 9-7
AddCartesianXMarker ... 9-8
AddCartesianYMarker ... 9-9
AddCartesianYMarkerToStack ... 9-10
AddDeltaMarker ... 9-10
AddMarker ... 9-11
AddNote ... 9-12
AddTraceCharacteristics ... 9-14
AddTraces ... 9-15
ApplyReportTemplate ... 9-17
ChangeProperty[ReportSetup] ... 9-17
ClearAllMarkers ... 9-24
ClearAllTraceCharacteristics ... 9-24
CloneReportsFromDatasetSolution ... 9-25
CopyPlotSettings ... 9-26
CopyReportDefinitions ... 9-26
CopyReportsData ... 9-27
CopyTraceDefinitions ... 9-27
CopyTracesData ... 9-28
CreateReport ... 9-29

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

CreateReportFromTemplate 9-41
CreateReportOfAllQuantities 9-42
DeleteMarker 9-43
DeleteAllReports 9-43
DeleteReports 9-44
DeleteTraceCharacteristics 9-44
DeleteTraces 9-45
DoesSupportTraceCharacteristics 9-46
DumpAllReportsData 9-47
EditCartesianXMarker 9-47
EditCartesianYMarker 9-48
EditMarker 9-49
ExportEyeMaskViolation 9-49
ExportImageToFile [Reporter] 9-50
ExportModelImageToFile 9-54
ExportModelMeshToFile 9-57
ExportPlot3DToFile [Reporter] 9-58
ExportReport 9-59
ExportReportDataToFile 9-60
ExportTableToFile 9-60
ExportToFile 9-61
ExportToFile [Reporter] 9-62
ExportUniformPointsToFile 9-63
GetAllCategories 9-65
GetAllQuantities 9-66
GetAllReportNames 9-66
GetAvailableDisplayTypes 9-67
GetAvailableReportTypes 9-67
GetAvailableSolutions 9-68
GetChildNames [Report Setup] 9-69

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

GetChildObject [Report Setup] 9-69
GetChildTypes [ReportSetup] 9-70
GetCurvePropServerName 9-70
GetDisplayType 9-71
GetDynLinkIntrinsicVariables 9-72
GetDynLinkQtyValueState 9-72
GetDynLinkTraces 9-73
GetDynLinkVariableValues 9-73
GetName 9-74
GetObjPath [Design] 9-75
GetPropertyValue 9-75
GetPropNames [Reporter] 9-77
GetPropValue [Report Setup] 9-77
GetQtyExpressionsForSourceTrace 9-78
GetReportTraceNames 9-78
GetReportSummaryForRegressionTesting 9-79
GetSolutionContexts 9-79
GetSolutionDataPerVariation 9-80
GetDataUnits 9-82
GetDesignVariableNames 9-83
GetDesignVariableUnits 9-84
GetDesignVariableValue 9-85
GetDesignVariationKey 9-86
GetImagDataValues 9-86
GetPerQuantityPrimarySweepValues 9-87
GetRealDataValues 9-88
GetSweepNames 9-89
GetSweepUnits 9-90
GetSweepValues 9-90
IsDataComplex 9-91

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

IsPerQuantityPrimarySweep 9-92
Release Data 9-93
GroupPlotCurvesByGroupingStrategy 9-93
ImportIntoReport 9-94
ImportReportDataIntoReport 9-95
MovePlotCurvesToGroup 9-95
MovePlotCurvesToNewGroup 9-96
OpenWindowForAllReports 9-97
OpenWindowForReports 9-97
PastePlotSettings 9-98
PasteReports 9-98
PasteReportsWithLegacyNames 9-99
PasteTraces 9-100
PasteTracesWithLegacyNames 9-100
RenameReport 9-101
RenameTrace 9-101
ResetPlotSettings 9-102
SavePlotSettingsAsDefault 9-103
SetLinkOutputTraces 9-103
SetPropValue [Report Setup] 9-104
UnGroupPlotCurvesInGroup 9-105
UpdateAllReports 9-105
UpdateReports 9-106
UpdateTraces 9-106
UpdateTracesContextAndSweeps 9-109
10 - Analysis Setup Module Script Commands 10-1
CopySetup 10-2
CopyDrivenSetup 10-2
CopyEigenSetup 10-3
CopySweep 10-3

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

DeleteDrivenSweep 10-4
DeleteSetups 10-5
DoesSweepSetupExists 10-5
EditCircuitSettings 10-6
EditDrivenSweep 10-6
EditFrequencySweep 10-8
EditSetup 10-9
ExportCircuit 10-20
GetSetupCount 10-24
GetSetups 10-25
GetSetups (Layout Editor) 10-25
GetSetupNames 10-26
GetSweepCount 10-26
GetSweeps 10-27
GetSweeps (Layout Editor) 10-27
InsertDrivenSweep 10-28
InsertFrequencySweep 10-32
InsertSetup 10-36
InsertSetup [Transient] 10-58
PasteDrivenSetup 10-60
PasteEigenSetup 10-61
PasteSetup 10-61
PasteSweep 10-62
RenameDrivenSweep 10-62
RenameSetup 10-63
ResetToTimeZero 10-63
RevertAllToInitial 10-64
RevertAllToZeroDisplacement 10-64
RevertSetupToInitial 10-65
RevertSetupToZeroDisplacement 10-65

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

SolveSetup 10-66
11 - Optimetrics Module Script Commands 11-1
CopySetup 11-7
DeleteSetups [Optimetrics] 11-8
DistributedAnalyzeSetup 11-8
EditSetup 11-9
EditSetup [Parametric] 11-20
EditSetup [Optimization] 11-21
EditSetup [Sensitivity] 11-25
EditSetup [Statistical] 11-28
EnableSetup 11-29
ExportDXConfigFile 11-30
ExportOptimetricsProfile 11-30
ExportOptimetricsResult 11-31
ExportParametricResults 11-32
ExportParametricSetupTable 11-33
ExportRespSurfaceMinMaxTable 11-33
ExportRespSurfaceRefinePoints 11-34
ExportRespSurfaceResponsePoints 11-35
ExportRespSurfaceVerificationPoints 11-35
GenerateVariationData [Parametric] 11-36
GetChildNames [Optimetrics] 11-36
GetChildObject [Optimetrics] 11-37
GetChildTypes [Optimetrics] 11-38
GetName 11-38
GetObjPath [Editor] 11-39
GetOptimetricResult 11-40
GetPropEvaluatedValue 11-40
GetPropNames [Optimetrics] 11-41
GetPropSIValue 11-42

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

GetPropValue [Optimetrics] 11-43
GetSetupNames [Optimetrics] 11-43
GetSetupNamesByType [Optimetrics] 11-44
ImportSetup 11-45
InsertSetup 11-46
InsertSetup [Parametric] 11-67
InsertSetup [Optimization] 11-71
InsertSetup [Sensitivity] 11-77
InsertSetup [Statistical] 11-79
PasteSetup [Optimetrics] 11-82
RenameSetup [Optimetrics] 11-82
SetPropValue [Optimetrics] 11-83
SolveAllSetup 11-84
SolveSetup [Optimetrics] 11-84
General Commands Recognized by the Optimetrics Module 11-85
CopySetup 11-87
DeleteSetups [Optimetrics] 11-88
DistributedAnalyzeSetup 11-88
EditSetup 11-89
EnableSetup 11-100
ExportDXConfigFile 11-101
ExportOptimetricsProfile 11-101
ExportOptimetricsResult 11-102
ExportParametricResults 11-103
ExportParametricSetupTable 11-104
ExportRespSurfaceMinMaxTable 11-104
ExportRespSurfaceRefinePoints 11-105
ExportRespSurfaceResponsePoints 11-106
ExportRespSurfaceVerificationPoints 11-106
ExportDOEResponseCurve 11-107

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

ExportDOEResponseCurveSlices 11-108
ExportDOEResponseSurface 11-108
ExportDOELocalSensitivity 11-109
ExportDOELocalSensitivityCurve 11-110
GetChildNames [Optimetrics] 11-110
GetChildObject [Optimetrics] 11-111
GetChildTypes [Optimetrics] 11-112
GetName 11-112
GetObjPath [Editor] 11-113
GetOptimetricResult 11-113
GetPropEvaluatedValue 11-114
GetPropNames [Optimetrics] 11-115
GetPropSIValue 11-115
GetPropValue [Optimetrics] 11-116
GetSetupNames [Optimetrics] 11-117
GetSetupNamesByType [Optimetrics] 11-118
ImportSetup 11-118
InsertSetup 11-119
PasteSetup [Optimetrics] 11-141
RenameSetup [Optimetrics] 11-141
SetPropValue [Optimetrics] 11-142
SolveAllSetup 11-143
SolveSetup [Optimetrics] 11-143
Parametric Script Commands 11-144
EditSetup [Parametric] 11-144
ExportParametricSetupTable 11-145
GenerateVariationData [Parametric] 11-146
InsertSetup [Parametric] 11-146
Optimization Script Commands 11-150
EditSetup [Optimization] 11-150

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] 11-154
Sensitivity Script Commands 11-159
EditSetup [Sensitivity] 11-159
InsertSetup [Sensitivity] 11-163
Statistical Script Commands 11-166
EditSetup [Statistical] 11-166
InsertSetup [Statistical] 11-167
12 - User-Defined Document Script Commands 12-1
AddDocument 12-1
DeleteAllDocuments 12-2
DeleteDocument 12-3
EditDocument 12-3
GetDocumentDefinitionNames 12-5
GetDocumentNames 12-5
RenameDocument 12-6
SaveHtmlDocumentAs 12-6
SavePdfDocumentAs 12-7
UpdateAllDocuments 12-8
UpdateDocument 12-8
ViewHtmlDocument 12-9
ViewPdfDocument 12-9
Example Python Script: Defining a Document 12-10
13 - User-Defined Solutions Commands 13-1
CreateUserDefinedSolution 13-1
DeleteUserDefinedSolutions 13-2
EditUserDefinedSolution 13-3
GetUserDefinedSolutionNames 13-4
GetUserDefinedSolutionProperties 13-4
14 - Network Data Explorer Script Commands 14-1
AddDiffPair 14-3

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

Cascade (SPISim) 14-4
ClearDiffPairs 14-5
Clone 14-6
Close 14-7
Combine (SPISim) 14-8
Deembed (SPISim) 14-9
DeembedBack (SPISim) 14-10
DeembedFront (SPISim) 14-11
DisableDiffPairs 14-12
EnableDiffPairs 14-13
ExportCitiFile 14-13
ExportFullWaveSpice 14-15
ExportMatlab 14-19
ExportNMFData 14-20
ExportNetworkData 14-20
ExportSpreadsheet 14-22
ExportTouchstone 14-24
ExportTouchstone2 14-27
Extract (SPISim) 14-29
GetFrequencies 14-30
GetFrequencyCount 14-30
GetName 14-31
GetPortCount 14-32
GetPortNumber 14-33
GetPostProcSettings 14-34
GetSolutionVariation 14-34
GetVariation 14-35
HasSameData 14-36
LoadSolution 14-38
Open 14-38

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

Rename (SPISim) 14-39
Renormalize (SPISim) 14-40
Reorder 14-41
Reorder (SPISim) 14-42
Reset 14-43
SetAllPortImpedances 14-44
SetPortDeembedDistance 14-45
SetPortImpedance 14-46
SetPostProcSettings 14-47
Smooth 14-48
Stretch (SPISim) 14-49
Terminate 14-50
15 - CompInstance Script Commands ..... (multiple)
Callback Scripting Using CompInstance Object ..... (multiple)
CompInstance Functions 54
GetComponentName 54
GetInstanceID [Component Instance] 55
GetInstanceName [Component Instance] 55
GetParentDesign 56
GetPropHost 56
GetPropServerName 57
16 - Schematic Scripting 16-1
Method Format 16-1
Editor Scripting IDs 16-4
Format of IDs for different schematic objects: 16-4
Format for Components 16-4
Create Method List 16-5
CreateArc (Schematic Editor) 16-7
CreateCircle (Schematic Editor) 16-9
CreateComponent (Schematic Editor) 16-11

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

CreateCurve [Schematic] ... 16-13
CreateGlobalPort (Schematic Editor) ... 16-15
CreateGround (Schematic Editor) ... 16-17
CreateLine (Schematic Editor) ... 16-18
CreateNPort (Schematic Editor) ... 16-20
CreatePagePort (Schematic Editor) ... 16-21
CreateIPort (Schematic Editor) ... 16-23
CreatePolygon (Schematic Editor) ... 16-25
CreateRectangle (Schematic Editor) ... 16-27
CreateText (Schematic Editor) ... 16-29
CreateWire (Schematic Editor) ... 16-31
General Method List ... 16-33
Activate (Schematic Editor) ... 16-35
AddPinGrounds (Schematic Editor) ... 16-36
AddPinIPorts (Schematic Editor) ... 16-38
AddPinPageConnectors (Schematic Editor) ... 16-39
AddPinCustomTerminations (Schematic Editor) ... 16-40
AlignHorizontal (Schematic Editor) ... 16-42
AlignVertical (Schematic Editor) ... 16-43
BringToFront (Schematic Editor) ... 16-44
CloseEditor (Schematic Editor) ... 16-45
Copy (Schematic Editor) ... 16-46
CopyData [Schematic Editor] ... 16-47
CopySubdesign [Schematic Editor] ... 16-47
CreatePage (Schematic Editor) ... 16-48
Cut (Schematic Editor) ... 16-49
DeactivateOpen (Schematic Editor) ... 16-50
DeactivateShort (Schematic Editor) ... 16-50
Delete (Schematic Editor) ... 16-51
DeletePage (Schematic Editor) ... 16-52

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

ElectricRuleCheck (Schematic Editor) 16-53
ExportImage (Schematic Editor) 16-54
FindElements (Schematic Editor) 16-55
FitToBorder [Schematic Editor] 16-57
FlipHorizontal (Schematic Editor) 16-58
FlipVertical (Schematic Editor) 16-59
GridSetup (Schematic Editor) 16-60
Move (Schematic Editor) 16-62
NameNets (Schematic Editor) 16-63
PageBorders (Schematic Editor) 16-64
Pan (Schematic Editor) 16-65
Paste (Schematic Editor) 16-65
PasteData [Schematic Editor] 16-66
PasteDesign (Schematic Editor) 16-67
PushExcitations 16-68
Rotate (Schematic Editor) 16-70
SelectAll (Schematic Editor) 16-72
SelectPage (Schematic Editor) 16-73
SendToBack 16-73
SetPageData [Schematic Editor] 16-74
ShowVariableBlock (Schematic Editor) 16-76
SortComponents (Schematic Editor) 16-77
Wire (Schematic Editor) 16-78
ZoomArea (Schematic Editor) 16-78
ZoomIn (Schematic Editor) 16-79
ZoomOut (Schematic Editor) 16-80
ZoomPrevious (Schematic Editor) 16-81
ZoomToFit (Schematic Editor) 16-81
Property Method List 16-82
ChangeProperty (Schematic Editor) 16-83

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

GetEvaluatedPropertyValue (Schematic Editor) ... 16-84
GetProperties (Schematic Editor) ... 16-84
GetPropertyAttribute [Schematic Editor] ... 16-86
GetPropertyValue (Schematic Editor) ... 16-87
SetPropertyValue (Schematic Editor) ... 16-87
Information Method List ... 16-88
GetAllPorts ... 16-89
GetCompInstanceFromRefDes (Schematic Editor) ... 16-89
GetComponentInfo (Schematic Editor) ... 16-90
GetComponentPins (Schematic Editor) ... 16-91
GetComponentPinInfo (Schematic Editor) ... 16-92
GetComponentPinLocation [Schematic Editor] ... 16-93
GetEditorName (Schematic Editor) ... 16-94
GetNetConnections (Schematic Editor) ... 16-94
GetNumPages [Schematic Editor] ... 16-95
GetPortInfo (Schematic Editor) ... 16-96
GetSelections (Schematic Editor) ... 16-97
GetSignals (Schematic Editor) ... 16-97
GetWireConnections (Schematic Editor) ... 16-99
GetWireInfo (Schematic Editor) ... 16-100
GetWireSegments (Schematic Editor) ... 16-102
17 - Nexxim Scripting ... 17-1
Nexxim Netlist Scripting ... 17-1
Analyze [Nexxim] ... 17-2
CopyEyeItemAsCommand ... 17-3
DeleteDesignInstance ... 17-3
EditImportData ... 17-4
EditNotes ... 17-5
ExportForSpice [Nexxim] ... 17-5
ExportForHSpice [Nexxim] ... 17-8

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

ExportNetlist [Nexxim] 17-10
GetModule [Nexxim] 17-10
GetName [Nexxim] 17-11
GetEditor [Nexxim] 17-12
GetResultsDirectory [Nexxim] 17-12
ImportData [Nexxim] 17-13
ImportDataFilePath [Nexxim] 17-16
InsertDesign [Nexxim] 17-17
PasteItemCommand 17-18
Redo [Nexxim] 17-18
RenameDesignInstance [Nexxim] 17-19
RenameImportData [Nexxim] 17-19
SetActiveEditor [Nexxim] 17-20
Undo [Nexxim] 17-21
UseCircuitSParameterDefinition 17-21
Nexxim Data Block Commands 17-22
AddTemperatureDataBlock 17-23
AddLibRefDataBlock 17-24
AddNetlistDataBlock 17-24
AddPrintToAuditDataBlock 17-25
AddStateVariableDataBlock 17-25
AddSubstrateDataBlock 17-26
AddDeviceNoiseDataBlock 17-26
EditTemperatureDataBlock 17-27
EditLibRefDataBlock 17-27
EditNetlistDataBlock 17-28
EditPrintToAuditDataBlock 17-29
EditStateVariableDataBlock 17-29
EditSubstrateDataBlock 17-30
EditDeviceNoiseDataBlock 17-30

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

GetTemperatureDataBlock 17-31
GetAllLibRefDataBlocks 17-31
GetAllNetlistDataBlocks 17-32
GetAllSubstrateDataBlocks 17-32
Remove [Nexxim] 17-33
Rename [Nexxim] 17-33
Nexxim Simulation Setup Commands 17-34
Add [Nexxim] 17-36
Add Alter Block [Nexxim] 17-37
Add AMI Analysis [Nexxim] 17-38
Add DC Analysis [Nexxim] 17-39
Add Envelope Setup [Nexxim] 17-41
Add HB 1-Tone [Nexxim] 17-46
Add HB N-Tone [Nexxim] 17-51
Add HSPICE Transient [Nexxim] 17-55
Add Linear Network Analysis [Nexxim] 17-57
Add Oscillator Analysis [Nexxim] 17-59
Add Oscillator N-Tone Analysis [Nexxim] 17-64
Add Oscillator Resonant Frequency Search [Nexxim] 17-69
Add PXF Setup [Nexxim] 17-74
AddQuickEyeAnalysis 17-79
Add System FD Analysis [Nexxim] 17-86
Add Transient Analysis [Nexxim] 17-87
Add TV Noise [Nexxim] 17-88
Add VerifEye Analysis 17-94
AddSweep [Nexxim] 17-100
Analyze All Nominal [Nexxim] 17-100
Analyze [Nexxim] 17-101
AnalyzeSweep [Nexxim] 17-101
Delete [Nexxim] 17-101

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

DisplayBiasPointInfo 17-102
DeleteSweep [Nexxim] 17-102
DynamicMeshOverlays 17-102
Edit [Nexxim] 17-103
Edit Alter Block [Nexxim] 17-103
Edit AMI Analysis [Nexxim] 17-104
Edit DC Analysis [Nexxim] 17-106
Edit Envelope Setup[Nexxim] 17-107
Edit HB 1-Tone [Nexxim] 17-113
Edit HB N-Tone [Nexxim] 17-118
Edit HSPICE Transient [Nexxim] 17-123
Edit Linear Network Analysis [Nexxim] 17-124
Edit Oscillator Analysis [Nexxim] 17-126
Edit Oscillator N-Tone Analysis [Nexxim] 17-131
Edit Oscillator Resonant Frequency Search [Nexxim] 17-136
Edit PXF Setup[Nexxim] 17-141
Edit QuickEye Analysis 17-147
Edit System FD Analysis [Nexxim] 17-153
Edit Transient [Nexxim] 17-154
Edit TV Noise [Nexxim] 17-156
Edit VerifEye Analysis 17-162
EditSweep [Nexxim] 17-168
GetAllSolutionSetups [Nexxim] 17-168
ListVariations [Nexxim] 17-168
RefreshMeshOverlays 17-169
RenameSweep [Nexxim] 17-169
Nexxim Linear Network Analysis 17-169
Add Analysis Options [Nexxim] 17-170
AddSimulationSetup [Nexxim] 17-171
Edit Analysis Options [Nexxim] 17-171

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

EditSimulationSetup [Nexxim] 17-172
ExportForSpice [Nexxim] 17-173
ExportForHSpice [Nexxim] 17-175
RemoveAnalysisOptions [Nexxim] 17-177
RemoveSimulationSetup [Nexxim] 17-177
RenameAnalysisOptions[Nexxim] 17-178
RenameSimulationSetup [Nexxim] 17-178
Interface Ports And Sources Commands 17-179
ChangePortProperty [Nexxim] 17-179
ChangeSourceProperty [Nexxim] 17-181
DeletePort [Nexxim] 17-182
DeleteSource [Nexxim] 17-182
GetAllPorts [Nexxim] 17-183
GetAllSources [Nexxim] 17-183
RenameSource [Nexxim] 17-184
Nexxim Component Manager Commands 17-184
ImportSandWComponent 17-184
ImportXparamComponent 17-185
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

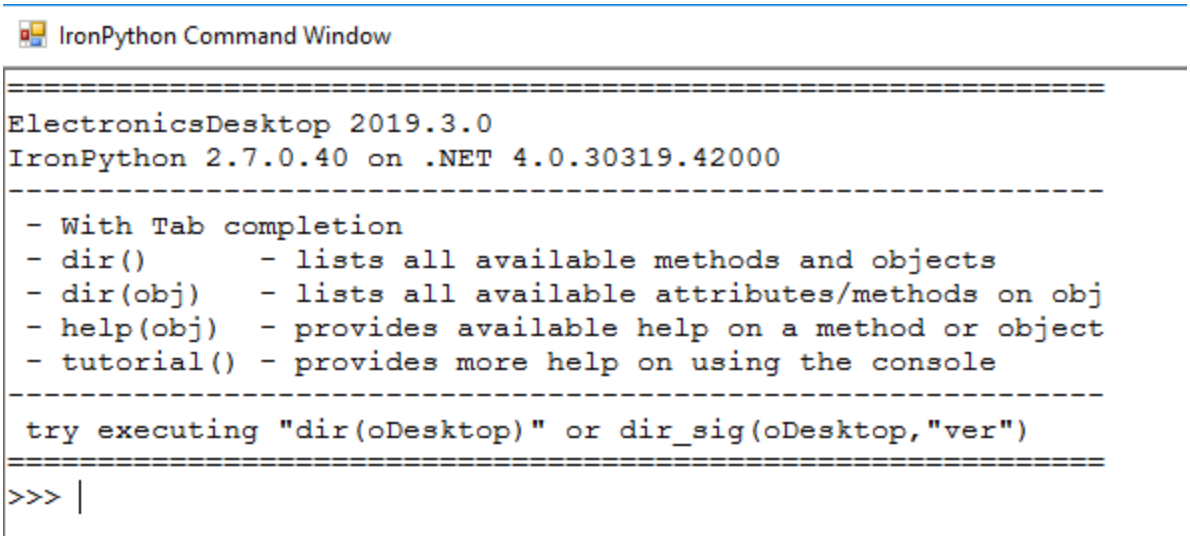
The advantages that IronPython use provides are significant:

- 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

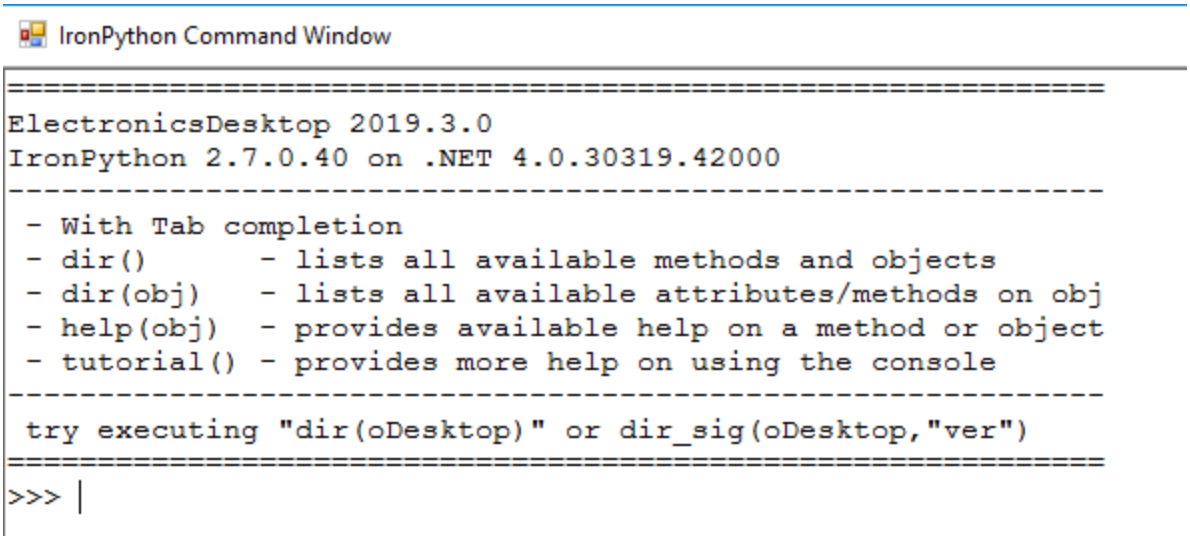
Appendix: IronPython Samples

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.
  - HFSS -runscript someScript.py to keep HFSS GUI open after completing script execution.
  - HFSS -features=beta -ng -runscriptandexit someScript.py to run HFSS in a non-graphical mode and exit after script completion. Note that this is a beta feature supported for the HFSS and 3D Layout design types.
- Register an IronPython script as an external tool using the **Tools** > **External Tools** menu item.

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

Script Argument for IronPython

Script Argument for IronPython

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

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

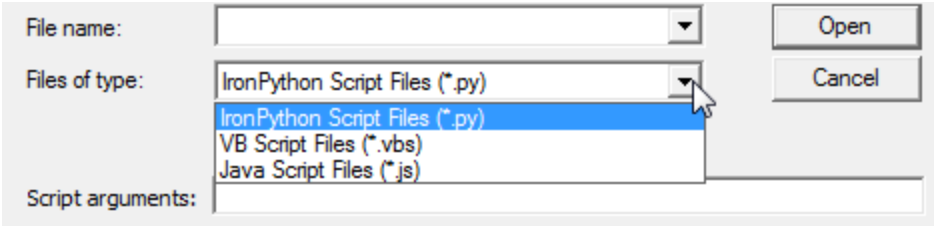


Figure 1: Run Script dialog and script arguments

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

Related Topics

IronPython Script Execution Environment

Scripting 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

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
```


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

```
# 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", _

 "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

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

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

Standalone IronPython

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

Running Standalone IronPython

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

It is located in:

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

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

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

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

Using a Recorded Script

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

Creating an External Script

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

- `import sys`

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

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

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

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

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

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

- `import ScriptEnv`

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

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

  or `ScriptEnv.InitializeNew(NonGraphical=True)`

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

You must end the script with:

- `ScriptEnv.Shutdown()`

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

Example Script

```
import sys

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

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


import ScriptEnv

ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")

oDesktop.RestoreWindow()

oProject = oDesktop.NewProject()

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

oDesign = oProject.SetActiveDesign("HFSSDesign1")

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

oEditor.CreateRectangle(

[

  "NAME:RectangleParameters",
```


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

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

IronPython Samples

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

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

Change property

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

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

Sample Script: ChangeProperty command to change color of a cone

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

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

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

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


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

```
   }]]
 ])
```

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

Create a Cone using IronPython

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

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


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

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

Sample Script: IronPython script to create a cone

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

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

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


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


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

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

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

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

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


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

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

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

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

Related Topics

Introduction to IronPython

Scripting Using Iron Python: Putting it all Together

Creating User Defined Primitives and User Defined Models in Python Scripts

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

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

Advantages Compared to C++

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

Changes compared to C

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

Structures

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

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

Return Values for UDM and UDP Functions

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

Constants

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

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

# with the type of the enum: UnitType

unitType = UnitType.LengthUnit
```

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

Methods

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

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

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

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

Output Parameters

The rule for the output parameters is as follows:

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

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

udmDefinition = udmFunctionLibrary.GetDefinition()


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

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

# Two output parameter; return value must be preserved

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

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

Comparison with C function:

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

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

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

'List Size' Parameters

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

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

Example:

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


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


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

Comparison with C function:

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

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

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

Added Parameters

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

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

Example:

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

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


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

Here cloneIdsSize is a new integer parameter.

Comparison with C function:

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

Developing a UDM/UDP

Creation

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

Location

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

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

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

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

Organize

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

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

For UDM:extension:

```
from UDM import *
```

For UDP:extension:

```
from UDP import *
```

Edit/Reload

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

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

UDPExtension

Import

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

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

```
from UDP import *
```

Main class: UDPExtension

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

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

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

IUDPExtension Methods

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

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

Mandatory Methods

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

GetLengthParameterUnits()

- returns string.

GetPrimitiveTypeInfo()

- returns UDPPrimitiveTypeInfo.

GetPrimitiveParametersDefinition2()

- returns a list of UDPPrimitiveParameterDefinition2 or None on failure

AreParameterValuesValid2(errorMsg, udpParams)

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

CreatePrimitive2(funcLib, udpParams)

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

Optional Methods

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

GetPrimitiveParameters()

- returns Python list of strings or NULL

GetRegisteredFaceNames()

- returns Python list of strings or NULL

GetRegisteredEdgeNames()

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

- returns Python list of strings or NULL

GetRegisteredVertexNames()

- returns Python list of strings or NULL

ProjectParametersOnToValidPlane2(currentUDPParams, projectedUDPParams)

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

MapParametersDefinitionVersions2(oldVersion, oldUDPParams)

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

GetOldPrimitiveParametersDefinition2(version)

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

Example UDP

```
import sys
class UDPExtension(IUDPExtension):

  def GetLengthParameterUnits(self):
    return "mm"

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


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

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

UDMExtension

Import

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

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

```
from UDM import *
```

Main class: UDMExtension

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

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

The class should implement IUDMExtension methods described below.

IUDMExtension Methods

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

Mandatory Methods

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

GetInfo()

- returns UDMInfo object populated with appropriate UDM information.

IsAttachedToExternalEditor()

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

CreateInstance(funcLib)

- funcLib is UDMFunctionLibrary
- returns UDMParameters.

GetUnits(instanceId)

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

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

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

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

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

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

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

  returns True on success, False on failure.

ReleaseInstance(instanceId)

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

GetAttribNameForEntityId()

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

GetAttribNameForPartId()

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

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

GetAttribNameForEntityId())

- Python UDMs should implement this method.

Optional Methods

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

DialogForDefinitionOptionsAndParams(self, defData, optData, params):

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

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

DialogForDefinitionAndOptions(self, defData, optData) [Deprecated]

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

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

GetInstanceSourceInfo(instanceId)

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

ShouldAttachDefinitionFilesToProject()

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

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

Example UDM

```
class UDMExtension(IUDMExtension):


  def IsAttachedToExternalEditor(self):
    return False


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


    return udmInfo
  ...
  ...
```

UDMFunctionLibrary

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

- CreateInstance
- Refresh

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

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

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

```
partId = 10

exact = True

udpPosition = UDPPosition(0,0,0)


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


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

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

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

Functions list:

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

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

Example:

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

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

UDM/UDP Functions

Return Values for Each UDM and UDP Function

ID – ID of created Object

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

Functions list:

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

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

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

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

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

UDP/UDM Structures and Constants

The following sections describe:

- UDP/UDM Structures
- UDP/UDM Constants

UDP/UDM Structures

Differences compared to C API:

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

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

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

  Example (creating UDMParameter list):

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

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

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

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

  Example:

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

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


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

- **UDP3x3Matrix**

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

  Example:

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

List of Structures

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

Example:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

UDP/UDM Constants

Full names of enum constants must be used in scripts.

Example:

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

Enum constants:

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

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

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

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

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

UDP Python Example

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

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


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

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


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


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

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

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

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

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

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

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


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

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

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

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

def GetPrimitiveTypeInfo(self):
    return primitive_info

def GetLengthParameterUnits(self):
    return length_units

def GetPrimitiveParametersDefinition2(self):
    return primitive_param_definitions

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

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


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

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

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

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

    """

    param_dict = self.get_param_dict(param_values)

    value = param_dict.get(param_name, None)

    return value


  def get_param_dict(self, param_values):

    """

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

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

    :return: dict of parameter name and values

    """

    udm_param_def = self.GetPrimitiveParametersDefinition2()

    param_dict = {}

    for i, param in enumerate(udm_param_def):

      param_value = param_values[i].Data

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

        param_dict[param.Name] = param_value

      else:

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

    return param_dict
```

Introduction to CPython

CPython can be used to:

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

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

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

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

Creating an External Script

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

```
import sys

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

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

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

Those lines are followed by:

```
import ScriptEnv

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

Follow that with:

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

Start ansysedt as GRPC server

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

-grpcsrv Flag

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

ansysedt –grpcsrv

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

ansysedt –grpcsrv portNumber

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

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

ansysedt –grpcsrv FirstPortNumber:LastPortNumber

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

ansysedt –grpcsrv FirstPortNumber:NumberOfPorts

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

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

Connect Functions:

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

Example:

```
import sys

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

import ScriptEnv

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

oProject = oDesktop.GetActiveProject()

print(oProject.GetName())
```

Launching Electronics Desktop

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

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

Where:

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

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

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

Connecting with a Running Instance of Electronics Desktop

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

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

Where:

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

Closing Electronics Desktop/Ending the Script

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

```
ScriptEnv.Shutdown()

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

-grpcsrv Flag

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

On Windows:

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

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

On Linux:

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

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

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

Related Topics:

Standalone Scripting in Iron Python

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

This page intentionally
left blank.

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

Ansys Electronics Desktop Scripting

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

Overview of Ansys Electronics Desktop Scripting Objects

Running a Script

Recording a Script

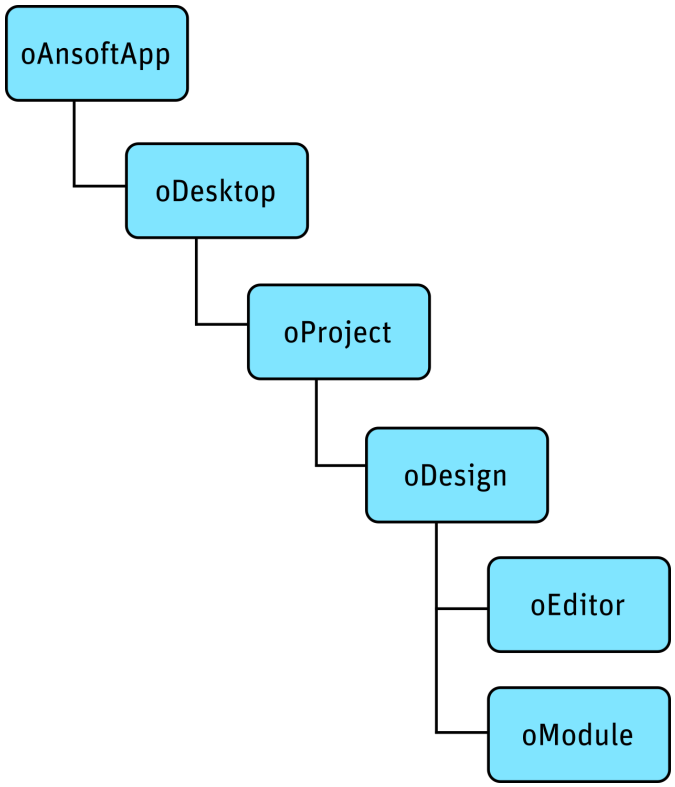
Working with Project Scripts

Executing a Script from Within a Script

Ansys Electronics Desktop Scripting Conventions

Overview of Electronics Desktop Scripting Objects

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



The commands are described below, followed by examples.

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

oAnsoftApp

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

In Iron Python, for example:

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

oDesktop

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

In Iron Python, for example:

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

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

- Desktop Object Script Commands.

oProject

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

In Iron Python, for example:

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

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

- Project Object Script Commands

oDesign

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

In Iron Python, for example:

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

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

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

oEditor

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

In Iron Python, for example:

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

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

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

- 3D Modeler Editor Script Commands
- Schematic 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
- Mesh Operations Module Script Commands
- Optimetrics 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")
```

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`.

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

In IronPython:

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

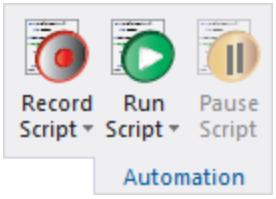
Running a Script

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

Within Electronics Desktop

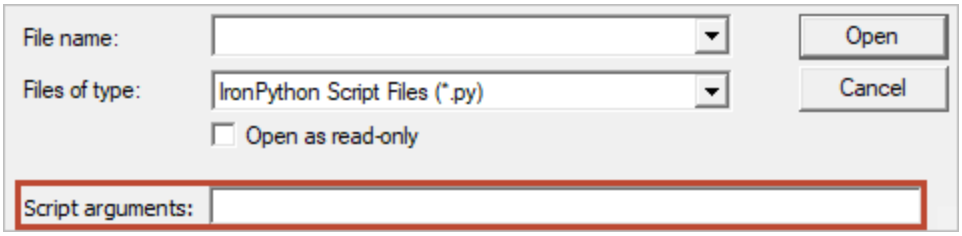
To run scripts in Electronics Desktop:

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.

   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.

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

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 Circuit help topic "Running Ansys Electronics Desktop from the Command Line".

Recording a Script

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

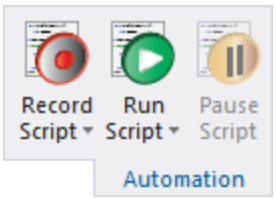
Scripts can be saved to an external file, or to the project.

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

Recording a Script to File

To record a script to file:

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



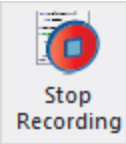
   A **Save As** file browser appears.

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

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

4. Click **Save**.

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



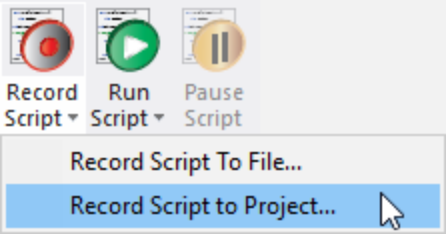
5. Perform the steps you want to record.
6. When you have finished recording the script, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

   The recorded script is saved to the folder you specified.

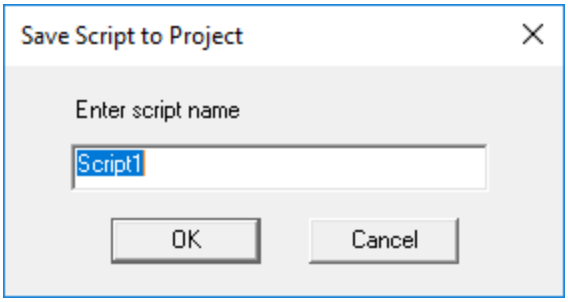
Recording a Script to a Project

To record a script to a project:

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



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



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

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

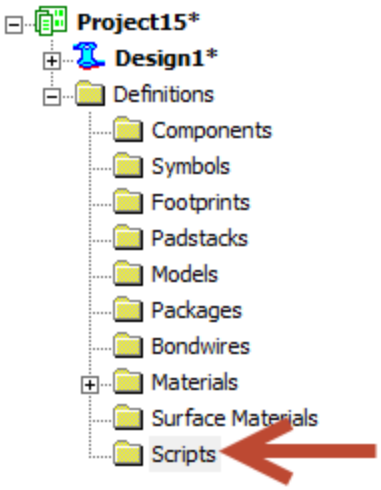
3. Perform the steps you want to record.
4. When you have finished, click **Stop Recording**, or select **Tools** > **Stop Script Recording**.

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

Working with Project Scripts

Scripts can be recorded to a project.

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



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



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

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

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

Executing a Script from Within a Script

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

In CPython:

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

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

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

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

Electronics Desktop Scripting Conventions

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

Consult the following topics:

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

Named Arguments

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

1. Named data, where name precedes data.

   For example:

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


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

2. Named Array, where name precedes array.

   For example:

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

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

   For example:

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

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

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

IronPython Example

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

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


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

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

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

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

Setting Numerical Values

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

- Pass in the number directly.

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

- Pass in a string containing the number with units.

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

- Pass in a variable name.

  ```
  var = 3.5

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

Layout Scripts and the Active Layer

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

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

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

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

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

Scripts and Locked Layers

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

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

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

This page intentionally
left blank.

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

2 - Object-Oriented Scripting

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

Object-Oriented Scripting Logic/Syntax

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

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

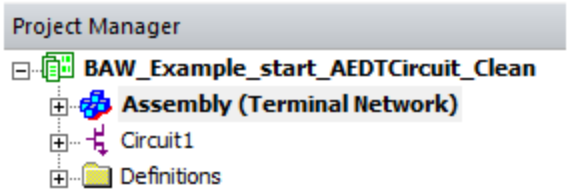
Basic Functions Overview

This section describes a typical object-oriented scripting workflow.

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

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

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



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

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

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

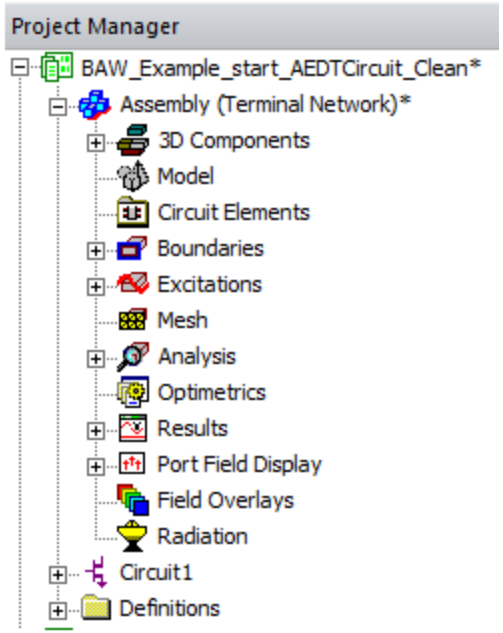
```
>>> oProject.GetChildNames()
['Assembly','Circuit1']
```

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

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

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

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

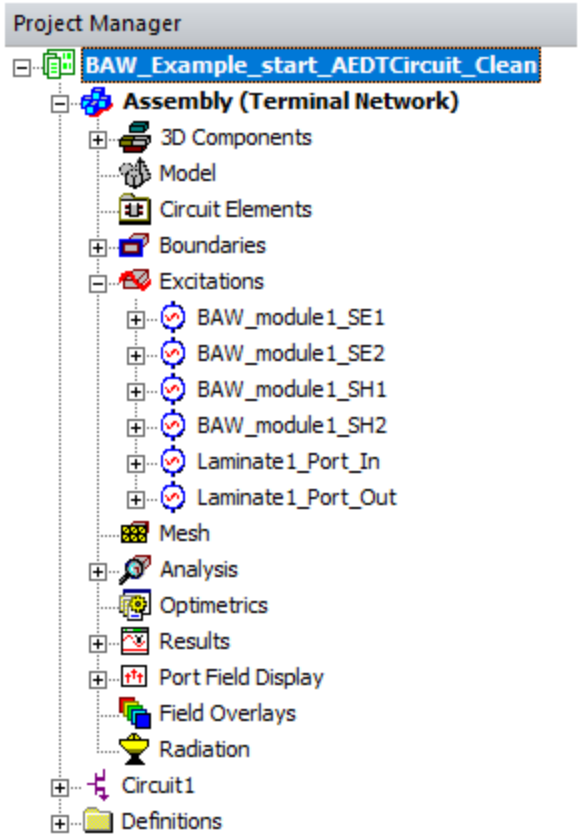


Any object in the **Project Manager** that has the '+' symbol for expansion is populated with child objects that can be queried. Once again, GetChildObject() can instance these objects. For example, set an instance for the Excitations:

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

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

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



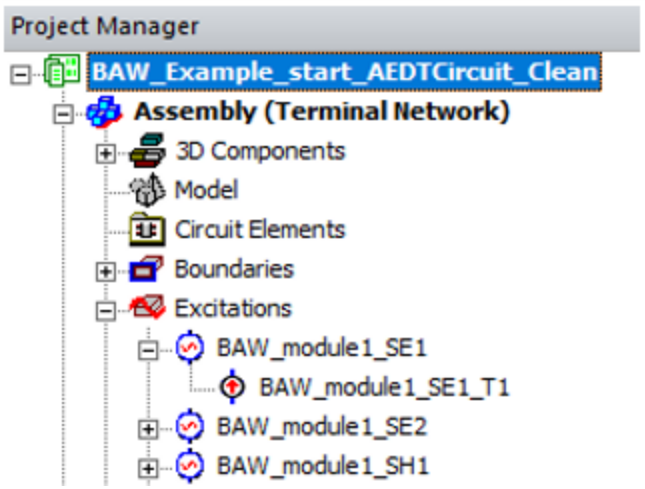
Those ports are outputted using GetChildNames():

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


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

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

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



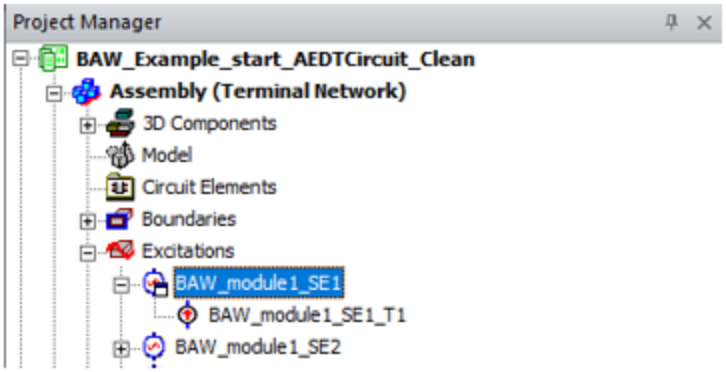
Through scripting, first define the Port Object (in this example, oPort) using the 'GetChildObject()' command for the first port, 'BAW_module1_SE1.' Then determine its Object Child Name, the terminal definition:

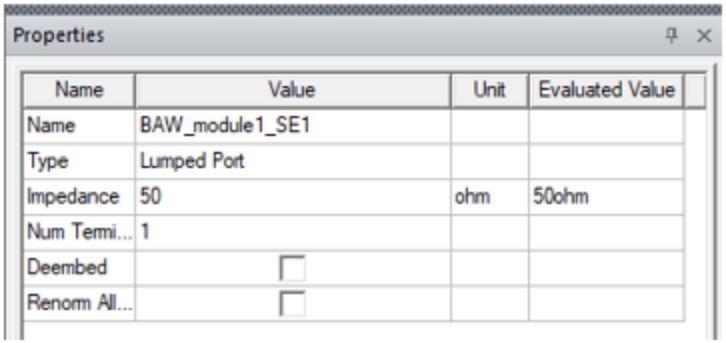
```
>>> oPort = oExcitations.GetChildObject('BAW_module1_SE1')
>>> oPort.GetChildNames()
['BAW_module1_SE1_T1']
```

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

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

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





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

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


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

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

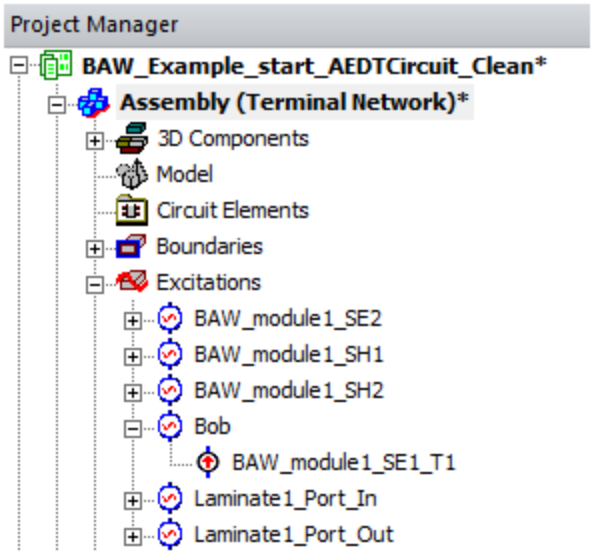
Once you identify the desired object and know the desired property, you can access the value via GetPropValue(). For example, if you want to retrieve the name of the object oPort:

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

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

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

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



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

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

Body Properties and Modification

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

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

Retrieving Variables

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

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


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

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

Retrieving Datasets and Values

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

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

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

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


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

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

GetSolutionData API

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

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

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

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

Summary

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

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

Each exposed script object supports the following COM functions:

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

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

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

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

  For example, an IcepakFEA design object has these children:

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


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

Four of the children are of “Module” type:

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

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

For example,

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

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

For example,

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

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

For example,

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

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

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

  For example,

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

- GetPropEvaluatedValue (<*PropName*>)

  For example,

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

- GetPropSIValue (<*PropName*>

  For example,

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

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

returns `"length / 2"`

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

returns `'0.4mm'`

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


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

returns `'0.0004'`

Additional Details Specific to AEDT Solvers

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

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

HFSS

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

HFSS 3D Layout

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

Maxwell 3D/2D

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

RMxprt

- Analysis, Field Overlays, Optimetrics, Results

Q3D

- Boundaries, Nets, Analysis, Optimetrics

2D Extractor

- Boundaries, Conductors, Analysis, Field Overlays, Optimetrics

Icepak

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

IcepakFEA

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

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

Circuit

- Optimetrics, Results

EMIT

- Coupling

Simplorer/Twin Builder

- Analysis, Optimetrics, Results

Additional Details on Boundaries/Excitations

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

These module script objects do not have properties

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

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

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

GetChildNames of these module object returns its immediate children

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

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

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


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

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

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

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

You can query/edit these properties

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

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

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

3D Component Encapsulation

These script interfaces are compliant with encapsulation. For example:

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

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

Additional Details on Solve Setup

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

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

This module script object has no property

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

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

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

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

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

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


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

Object-Oriented Scripting for Editors

Editors can be accessed via object-oriented scripting.

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

"Machine" is a child of RMxprt design.

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

Schematic Editor

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

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

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

Object-Oriented Scripting for Materials

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

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

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

Materials are child objects of the active project.

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

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

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

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

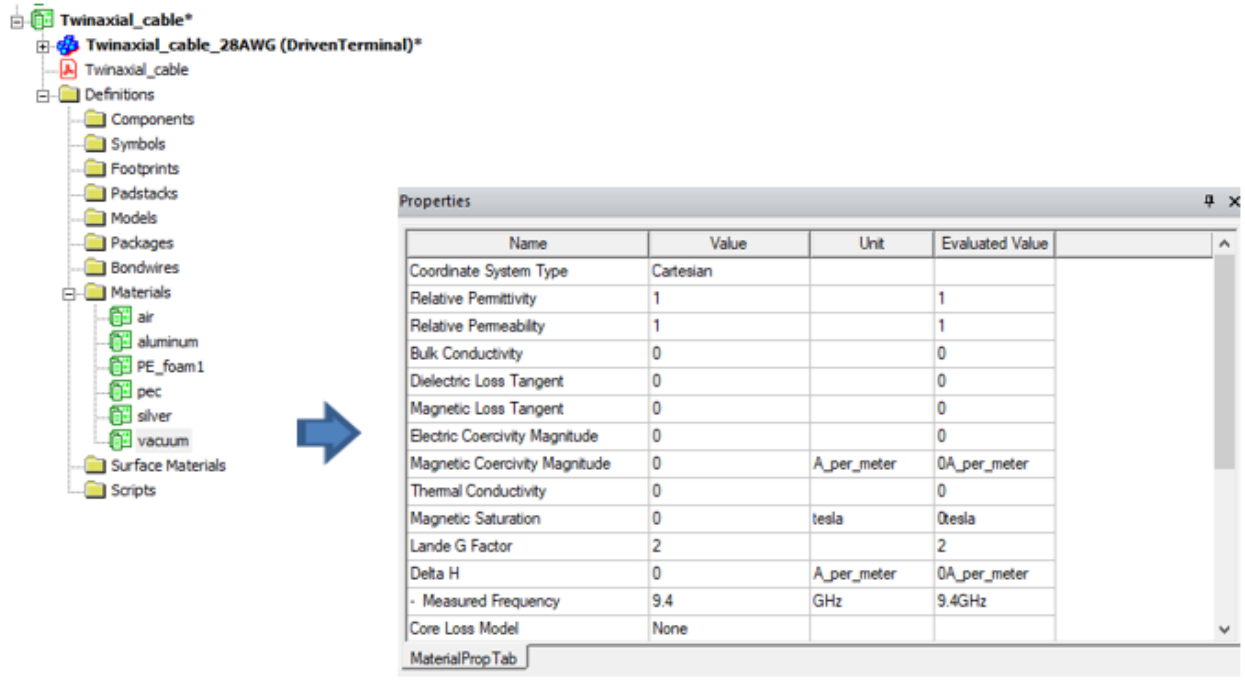
>>> omat.GetPropNames()

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

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

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

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



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.

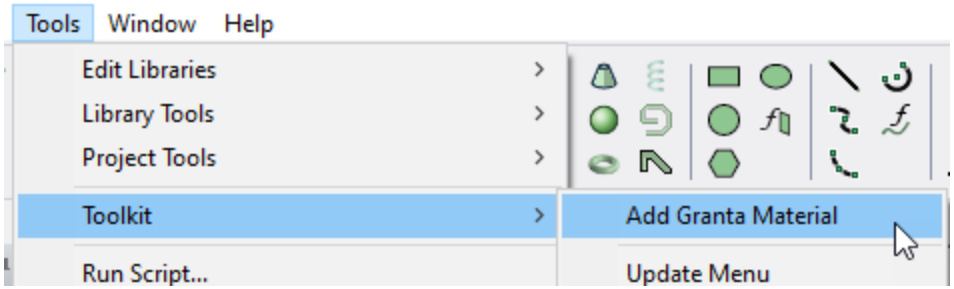
With this feature enabled you can then:

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

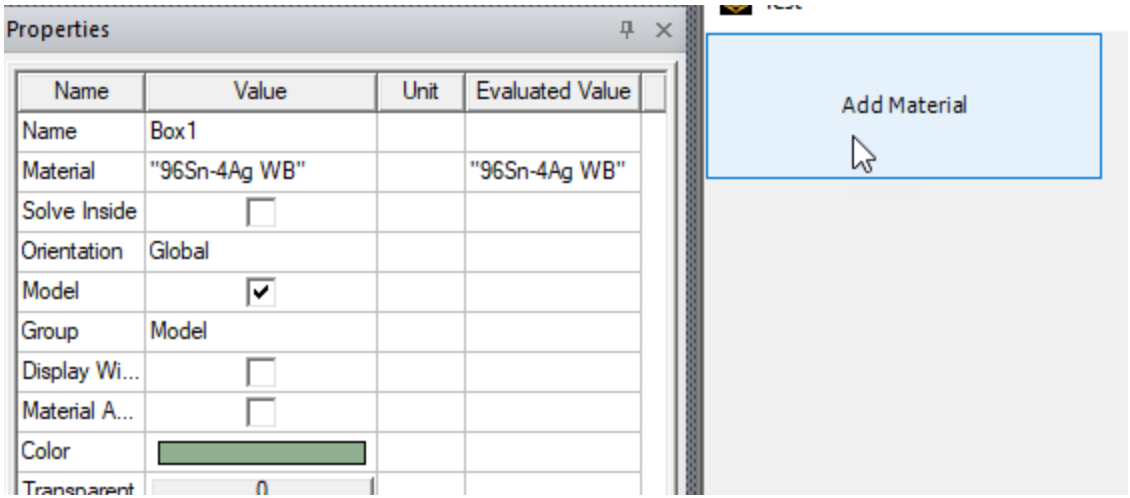
Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- 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, 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

PropHost Functions

Callback Scripting Using PropHost Object

GetArrayVariables

GetProperties

GetPropertyValue

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

GetVariables

GetVariableValue

SetPropertyValue

SetVariableValue

Example Use of Record Script and Edit Properties

Object Script Property Function Summary

Object Path

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

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

Property Object

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

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

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

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

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

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

Project Object

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

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

  oDesign = oProject("TeeModel")

  oVariable = oProject.GetChildObject("VariableName")

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

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

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

- SetPropValue(propertyPath, newValue)

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

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

Design Object

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

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

3D Modeler Object

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

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

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

Variable Object

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

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

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

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

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

Optimetrics Module Object:

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

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

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

Optimetrics Setup Object

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

oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')

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

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

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

ReportSetup(Results) Module Object:

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

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

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

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

ReportSetup(Results) Module Child Objects:

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

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

oRpt = oRptModule.GetChildObject(reportName)

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

oTrace = oRpt.GetChildObject(traceName)

oTrace = oRptModule.GetChildObject(ReportName/TraceName)

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

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

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

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.

Callback Scripting Using PropHost Object

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

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

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

Definitions

<**propName**> = text string

<**value**> = double

<**valueText**> = text string

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

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

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

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

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

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

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

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

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

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

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

The topics for this section include:

ChangeProperty for HFSS and Maxwell

ChangeProperty for Schematic and Layout

ChangeProperty (Schematic Editor and Layout Editor)

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

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

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

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

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

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

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

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

Property Array Structure

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

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


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

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

`<PropType>`

Type: string

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

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

`NewName`

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

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

`Description`

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

`NewRowPosition`

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

In HFSS, this only applies to the **Project** > **Project Variables** panel and the **HFSS** > **DesignProperties** panel.

In Maxwell, this only applies to the **Project** > **Project Variables** panel and the **Maxwell3D** > **DesignProperties**, **Maxwell2D** > **Design Properties**, or **RMxprt** > **Design Properties** panels. Specify the new zero-based row index of the variable or separator.

`ReadOnly`

Used to mark a property as "read only" so it cannot be modified. In HFSS and Maxwell, this flag can only be set for variables and separators.

`Hidden`

Used to hide a property so it cannot be viewed outside of the **Property** dialog box. In HFSS and Maxwell, this flag can only be set for variables and separators.

```
<PropTypeSpecificArgs>

  SeparatorProp: no arguments

  TextProp: "Value:=", <string>

  NumberProp: "Value:=", <double>

  ValueProp: "Value:=", <value>

  CheckboxProp: "Value:=", <bool>

  MenuProp: "Value:=", <string>
```


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

```
PointProp: "X:=", <double>, "Y:=", <double>

VPointProp: "X:=", <value>, "Y:=", <value>

V3DPointProp: "X:=", <value>, "Y:=", <value>, "Z:=", <value>

Material Button: "Material:=", <string>

Color Button: "R:=", <int>, "G:=", <int>, "B:=", <int>

Transparency Button: "Value:=", <double>
```

<PropTypeSpecificArgs> for VariableProps

Syntax:

```
"Value:=", <value>, <OptimizationFlagsArray>, <TuningFlagsArray>, <SensitivityFlagsArray>,
<StatisticsFlagsArray>
```

Parameters:

```
<OptimizationFlagsArray>

  ["NAME:Optimization",

  "Included:=", <bool>,

  "Min:=", <value>,

  "Max:=", <value>]

<TuningFlagsArray>

  ["NAME:Tuning",

  "Included:=", <bool>,

  "Step:=", <value>,

  "Min:=", <value>,
```


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

```
    "Max:=", <value>]
  <SensitivityFlagsArray>
    ["NAME:Sensitivity",
    "Included:=", <bool>,
    "Min:=", <value>,
    "Max:=", <value>,
    "IDisp:=", <value>]
  <StatisticsFlagsArray>
    ["NAME:Statistical",
    "Included:=", <bool>,
    "Dist:=", <Distribution>,
    "StdD:=", <value>,
    "Min:=", <value>,
    "Max:=", <value>,
    "Tol:=", <string>]
<Distribution>
```

Type: string

Value should be "Gaussian" or "Uniform"

`StdD`

Standard deviation.

`Min`

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

Low cut-off for the distribution.

`Max`

High cut-off for the distribution.

`Tol`

Tolerance for uniform distributions. Format is "<int>%". Example: "20%".

PropHost Functions

The following commands can be used to manipulate properties from a Property script.

GetEditor

GetPropHost

AddMenuProp

Use: Creates a new Menu property in tabType with name specified; choices are set to the values in choices; initial selection is initialChoice.

UI Access	NA		
Parameters	Name	Type	Description
	<tabType>	Integer	One of the following, where tab titles are shown in parentheses: 1 for PassedParameterTab ("Parameter Values") 2 for DefinitionParameterTab (Parameter Defaults") 3 for LocalVariableTab ("Variables" or "Local Variables") 4 for ProjectVariableTab ("Project variables") 5 for ConstantsTab ("Constants")

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

			6 for BaseElementTab ("Symbol" or "Footprint") 7 for ComponentTab ("General") 28 for CustomTab ("Intrinsic Variables") 39500 for Quantities ("Quantities") 39506 for Signals ("Signals") 39512 for Component ("Component")
	<propName>	String	Name of the new menu property
	<choices>	String	A comma-separated list of menu choice strings.
	<initialChoice>	String	One of the strings in the list of choices.
Return Value	None		

Python Syntax	AddMenuProp(<tabType>, <propName>, <choices>, <initialChoice>)
Python Example	`PropHost.AddProp(2,` `"ResChoices", "inline,upfront,parallel,series", "parallel")`

AddMenuProp2

Creates a new Menu property in tabTypeName with name specified; choices are set to the values in choices; initial selection is initialChoice.

UI Access	None		
Parameters	Name	Type	Description
	<tabType>	String	One of the following, where tab titles are shown in parentheses:

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

			PassedParameterTab ("Parameter Values") DefinitionParameterTab (Parameter Defaults") LocalVariableTab ("Variables" or "Local Variables") ProjectVariableTab ("Project variables") ConstantsTab ("Constants") BaseElementTab ("Symbol" or "Footprint") ComponentTab ("General") Component("Component") CustomTab ("Intrinsic Variables") Quantities ("Quantities") Signals ("Signals")
	<propName>	String	Name of the new menu property.
	<choices>	String	A comma-separated list of menu choice strings.
	<initialChoice>	String	One of the strings in the list of choices.
Return Value	None		

Python Syntax	AddMenuProp2(<tabTypeName>, <propName>, <choices>, <initialChoice>)
Python Example	`PropHost.AddMenuProp2("DefinitionParameterTab", "ResChoices", "inline,upfront,parallel,series", "parallel")`

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

AddProp

Use: Creates a new propType property in tabType with name and value specified.

<table>
<tr><td>UI Access</td><td colspan="3">NA</td></tr>
<tr><td rowspan="2">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><tabType></td><td>Integer</td><td>One of the following, where tab titles are shown in parentheses:
1 for PassedParameterTab ("Parameter Values")
2 for DefinitionParameterTab (Parameter Defaults")
3 for LocalVariableTab ("Variables" or "Local Variables")
4 for ProjectVariableTab ("Project variables")
5 for ConstantsTab ("Constants")
6 for BaseElementTab ("Symbol" or "Footprint")
7 for ComponentTab ("General")
28 for CustomTab ("Intrinsic Variables")
39500 for Quantities ("Quantities")
39506 for Signals ("Signals")
39512 for Component ("Component")</td></tr>
<tr><td></td><td><propType></td><td>Integer</td><td>One of the following to add new definition properties (e.g., to component, design or project):
0 for TextProp (any string)
1 for Menu (one,two,three)
2 for CheckboxProp ("true" or "false" values)</td></tr>
</table>

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

			3 for VariableProp (useful for Project and Local Variables, expression) 4 for VPointProp (x and y values, which can be expressions) 5 for V3DPointProp (x, y and z values, which can be expressions) 6 for NumberProp (simple number, no expression) 7 for ColorProp (takes an RGB value) 8 for PointProp (x and y values, no expressions) 11 for SeparatorProp (used for property list organization, the value is the separator "title") 13 for FileNameProp (special button property for filenames) 39506 for GenericProp (for generics in Twin Builder, SML and VHDL) 39500 for QuantityProp (for Twin Builder quantities, SML and VHDL) 39504 for SignalProp (for VHDL signals in Twin Builder) 39502 for VariablePropNU (like a `VariableProp`, but provides access to setting NetlistUnits in Component definition dialog for Designer; for Twin Builder, use `GenericProp` in `DefinitionParametersTab`, `QuantitiyProp` in `Quantities`, and `signalProp` in `Signals`)
	<propName>	String	Name of the new menu property.
	<valueText>	String	The property value
Return Value	None		

Python Syntax	AddProp(<tabType>, <propType>, <propName>, <valueText>)

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

Python Example	`PropHost.AddProp(2, 3, "W1", "10mm")`

Note:

AddProp adds a new property using the <propName> preceded by a separator "--". For instance, a new property added with the <propName> "xxx" will be added with the name "--xxx". Consequently, all subsequent script functions that wish to access the added property must now use the name "--xxx".

For example:

PropHost.AddProp 1, 11, "Time_Domain_Options",""

Prophost.SetHidden "--Time_Domain_Options",1

propType	Property Type	Value Example
0	Text	"hello"
1	Menu	"one,two,three"
2	Checkbox	"true" or "false"
3	Variable	"34mm" or "2*val1"
4	VPoint	"x1,y1"
5	V3DPoint	"x1,y1,z1"
6	Number	"17" or "22pF"
7	Color	
8	Point	"34,27"
9	Value	"34mm" or "2*val1"
11	Separator	"colors"

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

13	Filename	"c:\temp\ne22.s2p"

AddProp2

Creates a new propTypeName property in tabTypeName with name and value specified.

UI Access	NA		
Parameters	Name	Type	Description
	<tabType>	Type	One of the following, where tab titles are shown in parentheses: PassedParameterTab ("Parameter Values") DefinitionParameterTab (Parameter Defaults") LocalVariableTab ("Variables" or "Local Variables") ProjectVariableTab ("Project variables") ConstantsTab ("Constants") BaseElementTab ("Symbol" or "Footprint") ComponentTab ("General") Component("Component") CustomTab ("Intrinsic Variables") Quantities ("Quantities") Signals ("Signals")
	<propType>	String	One of the following to add new definition properties (e.g., to component, design or project):

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

			TextProp (any string) CheckboxProp ("true" or "false" values) VariableProp (useful for Project and Local Variables, expression) VPointProp (x and y values, which can be expressions) V3DPointProp (x, y and z values, which can be expressions) NumberProp (simple number, no expression) ColorProp (takes an RGB value) PointProp (x and y values, no expressions) SeparatorProp (used for property list organization, the value is the separator "title") FileNameProp (special button property for filenames) GenericProp (for generics in Twin Builder, SML and VHDL) QuantityProp (for Twin Builder quantities, SML and VHDL) SignalProp (for VHDL signals in Twin Builder) VariablePropNU (like a `VariableProp`, but provides access to setting NetlistUnits in Component definition dialog for Designer; for Twin Builder, use `GenericProp` in `DefinitionParametersTab`, `QuantitiyProp` in `Quantities`, and `signalProp` in `Signals`)
	<propName>	String	Name of the new menu property.
	<valueText>	String	The property value
Return Value	None		

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

Python Syntax	AddProp2(<tabTypeName>, <propTypeName>, <propName>, <valueText>)
Python Example	`PropHost.AddProp2("DefinitionParameterTab", "VariableProp", "W1", "10mm")`

ExecuteScript

Finds the named script in the Definitions/Scripts folder and runs that script; the script being run can also use the PropHost object.

UI Access	NA
Parameters	Name / Type / Description <scriptName> / String / Name of a script stored in a script library
Return Value	None

Python Syntax	ExecuteScript(<scriptName>)
Python Example	`PropHost.ExecuteScript("PropChangeScript")`

GetApplication

Returns the application object currently running the script.

UI Access	NA

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

Parameters	Name | Type | Description None | |
Return Value	Object The Application

Python Syntax	GetApplication()
Python Example	`oAnsoftApp2 = prophost.GetApplication()`

GetCallback

Finds named property and returns name of Callback script.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

Python Syntax	GetCallback(<propName>)
Python Example	`a = PropHost.GetCallback( "W1")`

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

GetChangedProperty

Use: If the script was called by a Callback associated with a property, this function returns the name of that property.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	String Returns "C" if the script was a Callback associated with the property named "C" and the script was called in response to the property "C" changing value.

Python Syntax	GetChangedProperty()
Python Example	`pn = PropHost.GetChangedProperty()`

GetDescription

Finds named property and returns description string

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	String

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

Python Syntax	GetDescription(<propName>)
Python Example	`a = PropHost.GetDescription( "W1")`

GetDesign

Use: Returns the interface to the specified design simulation.

Command: None

Syntax: GetDesign <DesignName>

Return Value: Interface to the specified design simulation.

Parameters: <DesignName>

Type: <string>

GetEditor

Use: Returns an interface to the editor requested IF the PropServer behind the PropHost is contained within that type of editor.

Command: None

Syntax: GetEditor(<editorName>)

Return Value: String

Python Syntax	GetEditor(<editorName>)

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

Python Example	`oEditor = GetEditor("SchematicEditor")`

GetEvaluatedText

Returns the evaluated value of the property, useful when the value is an expression.

UI Access	NA
Parameters	Name | Type | Description <propname> | String | The name of the property
Return Value	String The Evaluated Value

Python Syntax	GetEvaluatedText (<propname>)
Python Example	`ret = prophost.GetEvaluatedText("R")`

GetFileName

Finds the full path name to propName.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Full path file name or empty string.

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

Note:

Directory variables can be used in the property's value (e.g. $projectdir, $userlib, $syslib, $personallib) and these will be expanded to the correct path. GetFileName always returns a path string; if propName actually contains a variable expression, that expression is evaluated to a constant string before returning.

Python Syntax	GetFileName(<propName>)
Python Example	`a = PropHost.GetFileName("SubstrateFile")`

GetHidden

Finds named property and returns its Hidden flag.

UI Access	NA
Parameters	Name / Type / Description <propName> / String / Name of the property
Return Value	Returns 1 if property is hidden and 0 if it is not.

Python Syntax	GetHidden(<propName>)
Python Example	`a= PropHost.GetHidden( "W1")`

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

GetProgress

Use: Returns the percentage (from 0 to 100) of the simulation completed.

Command: None

Syntax: GetProgress(<simProgress>)

Return Value: String

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Return Value: Returns interface to PropHost.

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

GetPropServers

Returns array of objects that have properties showing on tabTypeName.

UI Access	NA
Parameters	Name | Type | Description None | |

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 propserver names.

Python Syntax	GetPropServers(<tabTypeName>)
Python Example	`objects = PropHost.GetPropServers("PassedParameterTab")` Returns array containing PropServers that have properties shown on PassedParameterTab; this would include only components and designs; individual properties can be accessed using standard notation, e.g. objects(0) might contain "CompInst@CAP_;1".

GetPropTabType

Finds named property and returns the id of the tab it is in.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Returns string

Python Syntax	GetPropTabType(<propName>)
Python Example	`a = PropHost.GetPropTabType( "W1")`

GetReadOnly

Finds named property and returns its ReadOnly flag.

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

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr></table>
Return Value	Returns 1 if property is read-only and 0 if it is not.

Python Syntax	GetReadOnly(<propName>)
Python Example	`a = PropHost.GetReadOnly( "W1");` `returns 1`

GetRunStatus

Use: Returns the status number of the specified design simulation.

Command: None

Syntax: GetRunStatus(<statusNumber>)

Return Value: Returns string.

GetTabTypeName

Finds named property and returns the name of the tab it is on.

UI Access	NA

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

Parameters	<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

Python Syntax	GetTabTypeName(<propName>)
Python Example	`a = PropHost.GetTabTypeName( "W1")`

GetText

Finds property in any tab and returns its value as a text string.

UI Access	NA
Parameters	<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

Python Syntax	GetText(<propName>)
Python Example	`a = PropHost.GetText("C")`

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

GetValue

Use: Finds property in any tab and returns its value as a double.

Command: None

Syntax: GetValue(<propName>)

Return Value: Returns double.

> **Note:**
>
> Values are returned in SI units. Compound SI units are, in general, not supported. Temperature values are returned in Celsius

IsValueConstant

Determines whether the specified property is a constant or not. Returns True if propName is evaluated to be a constant; returns False if propName is evaluated to be an expression.

UI Access	NA
Parameters	Name | Type | Description <propname> | String | The name of the property.
Return Value	Integer; non-zero for constant, zero for variable or expression.

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

Python Syntax	IsValueConstant (<propname>)
Python Example	`ret = prophost.IsValueConstant("R")`

PropertyExists

Finds named property and returns its property type.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property
Return Value	Returns 1 if property exists in any tab, 0 if it does not.

Python Syntax	PropertyExists(<propName>)
Python Example	`a = PropHost.PropertyExists( "W1")`

RemoveProp

Removes the named property from whichever tab it is found.

UI Access	NA
Parameters	Name | Type | Description <propName> | String | Name of the property

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

Return Value	None

Python Syntax	RemoveProp(<propName>)
Python Example	`PropHost.RemoveProp("W1")`

SetCallback

Finds named property and sets its Callback script.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<scriptName>	String	Name of a script stored in a script library.
Return Value	None		

Python Syntax	SetCallback(<propName>, <scriptName>)
Python Example	`PropHost.SetCallback("W1", "SynchronizeResistors")`

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

SetDescription

Finds named property and sets its description text.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<description>	String	Descriptive text.
Return Value	None		

Python Syntax	SetDescription(<propName>, <description>)
Python Example	`PropHost.SetDescription("W1", "this is the width of the gate")`

SetHidden

Finds named property and sets its Hidden flag.

UI Access	NA		
Parameters	Name	Type	Description
	<propName>	String	Name of the property
	<hidden>	Boolean	True or 1, false or 0.
Return Value	None		

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

Python Syntax	SetHidden(<propName>, <hidden>)
Python Example	`PropHost.SetHidden("W1", 1)`

SetReadOnly

Finds named property and sets its ReadOnly flag.

UI Access	NA
Parameters	Name / Type / Description <propName> / String / Name of the property <hidden> / Boolean / True or 1, false or 0.
Return Value	None

Python Syntax	SetReadOnly(<propName>, <hidden>)
Python Example	`PropHost.SetReadOnly("W1", 1)`

SetText

Finds property in any tab and sets its value to a text string.

UI Access	NA

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

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr><tr><td><text></td><td>String</td><td>The value for the property.</td></tr></table>
Return Value	None

Python Syntax	SetText(<propName>, <text>)
Python Example	`PropHost.SetText("C","22nF")`

SetValue

Finds property in any tab and sets its value to a double.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><propName></td><td>String</td><td>Name of the property</td></tr><tr><td><value></td><td>Real</td><td>A real number for the property value.</td></tr></table>
Return Value	None

Python Syntax	SetValue(<propName>, <value>)
Python Example	`PropHost.SetValue("C",2e-9)`

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

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()`

GetProperties

Gets a list of all the properties belonging to a specific <PropServer> and <PropTab>. This can be executed by the oProject, oDesign, or oEditor variables.

UI Access	N/A
Parameters	Name: *<PropTab>* Type: String Description: One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables")

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

			• ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<PropServer>	String	An object identifier, generally returned from another script method, such as Comp‡nst@R;2;3
Return Value	Array of strings containing the names of the appropriate properties.		

Python Syntax	GetProperties(<*PropTab*>, <*PropServer*>)
Python Example	`oEditor.GetProperties('PassedParameterTab', 'k')`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

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

UI Access	N/A

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

<table>
<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:`
	`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.

	`...`

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: <*VarName*> — Type: String — Description: 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")`

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	See table below

Name	Type	Description
<*propTab*>	String	One of the following, where tab titles are shown in parentheses: • PassedParameterTab ("Parameter Values") • DefinitionParameterTab (Parameter Defaults") • LocalVariableTab ("Variables" or "Local Variables") • ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General")

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

			• Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<*propServer*>	String	An object identifier, generally returned from another script method, such as Comp⊦nst@R;2;3
	<*propName*>	String	Name of the property.
	<*propValue*>	String	The value for the property
Return Value	None.		

Python Syntax	SetPropertyValue(<*propTab*>, <*propServer*>, <*propName*>, <*propValue*>)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

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.		

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

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

DeleteProject

DeleteRegistryEntry

DoesRegistryValueExist

DownloadJobResults

EnableAutoSave

ExportOptionsFiles

GetActiveProject

GetActiveScheduler

GetActiveSchedulerInfo

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

GetAutoSaveEnabled

GetBuildDateTimeString

GetCustomMenuSet

GetDefaultUnit

GetDesktopConfiguration

GetDistributedAnalysisMachines

GetDistributedAnalysisMachinesForDesignType

GetExeDir

GetGDIObjectCount

GetLibraryDirectory

GetLocalizationHelper

GetMessages

GetMonitorData

GetPersonalLibDirectory

GetProcessID

GetProjectDirectory

GetProjectList

GetProjects

GetRegistryInt

GetRegistryString

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

GetRunningInstancesMgr

GetSchematicEnvironment

GetScriptingToolsHelper

GetSysLibDirectory

GetTempDirectory

GetUserLibDirectory

GetVersion

IsFeatureEnabled

KeepDesktopResponsive

LaunchJobMonitor

NewProject

OpenAndConvertProject

OpenProject

OpenProjectWithConversion

PageSetup

PauseRecording

PauseScript

Print

QuitApplication

RefreshJobMonitor

ResetLogging

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

RestoreProjectArchive

RestoreWindow

ResumeRecording

RunACTWizardScript

RunProgram

RunScript

RunScriptWithArguments

SelectScheduler

SetActiveProject

SetActiveProjectByPath

SetCustomMenuSet

SetDesktopConfiguration

SetLibraryDirectory

SetProjectDirectory

SetRegistryFromFile

SetRegistryInt

SetRegistryString

SetSchematicEnvironment

SetTempDirectory

ShowDockingWindow

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

Sleep

StopSimulations

SubmitJob

TileWindows

Related Topics:

Desktop Commands For Registry Values

ImportExport Tool Commands

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

AddMessage

Add a message with severity and context to message window.

UI Access	N/A
Parameters	

Name	Type	Description
<*projectName*>	String	Project name. Passing an empty string adds the message as the desktop global message.
<*designName*>	String	Design name. Ignored if project name is empty. Passing an empty string adds the message to project node in the message tree.
<*severity*>	Integer	One of "Error", "Warning" or "Info". Anything other than the first two is treated as "Info" 0 = Informational, 1 = Warning, 2 = Error, 3 = Fatal
<*msg*>	String	The message for the message window.
<*category*>	String	Optional. The category is created with the message under the design tree node if the category does not exist. If the category already exists, the new message is added to the end of the existing category. It is ignored if the project or design is empty. If missing or empty, the message is added to the Design node in the message tree.

Return Value	None.

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

UI Access	Select Scheduler
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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>
Return Value	A Boolean result about download complete or not

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

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/Circuit')`

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 ("Circuit")`

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", "Circuit1", 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/Circuit/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/Circuit')`

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.

OpenAndConvertProject

Opens a legacy project and converts or copies it to .aedt format.

UI Access	Click **File** > **Open**, and choose a legacy project
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*itemPath*></td><td>String</td><td>full project path of the legacy project</td></tr><tr><td><*legacyChoice*></td><td>Integer</td><td>0: show conversion dialog box, (same as **File** > **Open** of a legacy file) 1: rename (changes extension to .aedt, the original file and results are renamed) 2: copy (creates new file with .aedt extension, and the original file and results remain available)</td></tr></table>
Return Value	An object reference to the newly opened project which has the .aedt extension

Warning:

If project file/results with the same name and .aedt extension already exist in the same directory, they will be overwritten.

Python Syntax	OpenAndConvertProject(*filePath*, *legacyChoice*)

Python Example	`oProject = oDesktop.OpenAndConvertProject("c:\files\optimtee.hfss", 1)`
	Note: optimtee.hfss is gone after this code executes
	`oProject = oDesktop.OpenAndConvertProject("c:\files\optimtee.hfss", 2)`
	Note: optimtee.hfss remains after this code executes

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.

PageSetup (Layout Editor)

Use: Specifies page setup for printing.

Command: **File>Page Setup**

Syntax: PageSetup <ArgArray>

Return Value: None.

Parameters: <Margins>: Page margins in implicit units of inches.

<Border>: Integer value indicating to draw border (1) or not to draw (0).

<DesignVars>: Integer value indicating to draw design vars (1) or not to draw (0).

PauseRecording

Temporarily stop script recording.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	None

Python Syntax	PauseRecording()

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.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

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.

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

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

QuitApplication

Exits the desktop.

UI Access	**File** > **Exit**.
Parameters	None.
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.

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

Return Value	A 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"

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.		

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

Python Syntax	ResetLogging(<*logFile*>, <*logLevel*>)
Python Example	`oDesktop.ResetLogging("C:/Project1.aedtresults/", 1)`

RestoreProjectArchive

Restores a previously archived project to a specified path.

UI Access	**File** > **Restore Archive**.
Parameters	

Name	Type	Description
<*ArchiveFilePath*>	String	Path to archived file
<*ProjectFilePath*>	String	Path to restore location
<*OverwriteExistingFiles*>	Boolean	True to overwrite an existing file of the same name; else False.
<*OpenProjectAfterRestore*>	Boolean	True to open the project after it is restored; else False.

Return Value	None.

Python Syntax	RestoreProjectArchive (<*Archivefilepath*>, <*ProjectFilePath*>, <*OverwriteExistingFiles*>, <*OpenProjectAfterRestore*>)
Python Example	`oDesktop.RestoreProjectArchive("C:\Users\jdoe\Documents\OptimTee.aedtz", "C:\Documents\OptimTee.aedt", False, True)`

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

RestoreWindow

Restores a minimized Desktop window.

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

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

ResumeRecording

Resume recording a script.

UI Access	N/A
Parameters	None.
Return Value	None

Python Syntax	ResumeRecording()

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.ResumeRecording()`

RunACTWizardScript

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

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

RunProgram

Runs an external program.

UI Access	NA		
Parameters	Name	Type	Description
	<ProgName>	String	Name of the program to run.
	<ProgPath>	String	Location of the program. Pass in an empty string to use the system path.
	<WorkPath>	String	Working directory in which program will start.
	<ArgArray>	Array of Strings	Arguments to pass to the program. If no arguments, pass in `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	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	See table below

Name	Type	Description
<*SCriptPath*>	String	Name or full path of the script to execute. If the full path to the script is not specified, Twin Builder searches for the specified script in the following locations, in this order: 1. Personal library directory. This is the **PersonalLib** subdirectory in the project directory. The project directory can be specified in the **General Options** dialog box (click >**Tools** > **Options** > **General Options** to open this dialog box) under the **Project Options** tab. 2. User library directory. This is the **userlib** subdirectory in the library directory. The library directory can be specified **General Options** dialog in the box (click **Tools** > **Options** > **General Options** to to open this dialog box) under the **Project Options** tab.

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

			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")`

RunScriptWithArguments

Similar to RunScript, the command launches another script from within the currently executing script, but with arguments.

UI Access	NA		
Parameters	Name	Type	Description
	<*ScriptPath*>	String	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.

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

			• 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
	<*Arguments*>	String	The arguments to supply to the specified script.
Return Value	Return code (in long format) for the script method.		

Python Syntax	RunScriptWithArguments (<*ScriptPath*>, <*Arguments*>)
Python Example	`oDesktop.RunScriptWithArguments` `("C:/Project/test2.py", "foo")`

SelectScheduler

Selects the scheduler used for batch job submission. It tries non-graphical selection of the scheduler, attempting to get version information from the scheduler in order to check for successful selection. If unable to get the information, it displays the **Select Scheduler** window and waits for the user to complete the settings.

UI Access	**Tools** > **Job Management** > **Select Scheduler**.		
Parameters	Name	Type	Description
	<*option*>	String	One of the following options (not case sensitive): • Empty string for remote RSM service

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

			• "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 password 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")`

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

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")`

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.

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

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*customMenuSet*></td><td>String</td><td>Name of desired menu set. Can be one of: 'Default' 'EM' 'RF' 'RF.0' 'SI' 'SI1.0' 'SI2.0' 'Twin Builder' </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	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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*configName*></td><td>String</td><td>Name of desired Desktop configuration. Can be one of: 'All' 'EM' 'RF' 'SI' 'Twin Builder' </td></tr></table>
Return Value	None

Python Syntax	SetDesktopConfiguration (<*configName*>)
Python Example	`oDesktop.SetDesktopConfiguration('RF')`

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

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")`

SetProjectDirectory

Sets the project directory path.

UI Access	N/A
Parameters	Name | Type | Description <*DirectoryPath*> | String | The path to the project directory. This should be writeable by the user.
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	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	Name | Type | Description <*filePath*> | String | 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.

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

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

	<*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/Circuit/UpdateReportsDynamicallyOnEdits', 0)`

SetRegistryString

Sets registry key to a string value.

UI Access	N/A		
Parameters	Name	Type	Description
	<*KeyName*>	String	Full name of registry key, including path.
	<*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*>)

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.SetRegistryString('Desktop/ActiveDSOConfigurations/Circuit', '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><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*schEnv*></td><td>Integer</td><td>Desired schematic environment. Can be one of: 0 (Circuit) 1 (Twin Builder) 2 (Maxwell Circuit) </td></tr></table>
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.

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 <*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")`

ShowDockingWindow

Shows or hides a docking window.

UI Access	Right click docking window > Show/Hide.
Parameters	Name | Type | Description <*windowName*> | String | The window name (for example, "Message Manager", "Component Libraries", "Properties") <*show*> | Boolean | True to show; False to hide.
Return Value	None.

Python Syntax	ShowDockingWindow (<*windowName*>, <show>)

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.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	Name: *<TimeInMilliseconds>* — Type: Integer — Description: The time that the execution should be suspended in milliseconds
Return Value	None

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**.

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><*settingsPath*></td><td>String</td><td>Path to the settings file (exported from the Submit Job dialog box) to use for submission.</td></tr><tr><td><*projectPath*></td><td>String</td><td>Path to the project file to use in the batch job. This could be an archive (.aedtz file) or an un-archived project.</td></tr><tr><td><*design*>*(optional)*</td><td>String</td><td>Name of the design to use for batch solve.</td></tr><tr><td><*setup*>*(optional)*</td><td>String</td><td>Name of the specific setup to solve.</td></tr></table>
Return Value	Array of job ID strings (empty if no jobs submitted).

Python Syntax	SubmitJob(<*settingsPath*>, <*projectPath*>, <*design*>, <*setup*>)

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

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 main menu, **Window** >**Tile Horizontally** or **Window** >**Tile Vertically**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*TilingFlag*></td><td>Integer</td><td>• 0 – Tile vertically. • 1 – Tile horizontally.</td></tr></table>
Return Value	None.

Python Syntax	TileWindows(<*TilingFlag*>)
Python Example	`oDesktop.TileWindows(0)`

Desktop Commands For Registry Values

The Ansys Registry is stored as XML format file. By default it is located at C:\Users\<UserName>\Documents\Ansoft\<AnsysProductNameversion>\config\<PC_NAME>_user.XML. Most of the Ansys product configuration information is stored in this XML file. These methods allow you to change the product configuration.

For example, to set the DSO & HPC analysis setup for HFSS using a Python script:

1. Start Circuit.
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")
```


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

```
# Set Active Setup to "test"
oDesktop.SetRegistryString("Desktop/ActiveDSOConfigurations/Circuit", "test")
```

See the following subtopics:

DeleteRegistryEntry

DoesRegistryValueExist

GetRegistryInt

GetRegistryString

SetRegistryFromFile

SetRegistryInt

SetRegistryString

DeleteRegistryEntry

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

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*pathToRegistry*></td><td>String</td><td>Path to Registry entry.</td></tr></table>
Return Value	Boolean: • **True** – Key has been deleted. • **False** – Key does not exist.

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

Python Syntax	DeleteRegistryEntry(<*pathToRegistry*>)
Python Example	`res = oDesktop.DeleteRegistryEntry("Desktop/ColorScheme")`

DoesRegistryValueExist

Determines whether a registry value exists.

UI Access	N/A
Parameters	Name | Type | Description <*KeyName*> | String | Full name of registry key, including path.
Return Value	Boolean: • **True** – Key exists. • **False** – Key does not exist.

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

GetRegistryInt

Obtains registry key integer value.

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 <*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/Circuit/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.

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

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

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 | Type | Description <*filePath*> | String | 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.

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

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

	<*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/Circuit/UpdateReportsDynamicallyOnEdits', 0)`

SetRegistryString

Sets registry key to a string value.

UI Access	N/A		
Parameters	Name	Type	Description
	<*KeyName*>	String	Full name of registry key, including path.
	<*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*>)

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.SetRegistryString('Desktop/ActiveDSOConfigurations/Circuit', 'Local')`

ImportExport Tool Commands

These oDesktop commands are run by using the GetTool script to call the ImportExport Tool.

```
oTool = oDesktop.GetTool("ImportExport")
```

Scripts run via the ImportExport Tool include:

ImportANF

ImportANFV2

ImportAutoCAD

ImportAWRMicrowaveOffice

ImportEDB

ImportExtracta

ImportGDSII

ImportGerber

ImportIDF

ImportIDFandMerge

ImportIPC

ImportODB

ImportXFL

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

ImportANF

Imports an ANF file into a new project. For older ANFv2 files, use ImportANFv2.

UI Access	**File** > **Import** > **ANF**.		
Parameters	Name	Type	Description
	<ANFfilename>	String	Full path of ANF file.
Return Value	None.		

Python Syntax	ImportANF(*<ANFfilename>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportANF('C:\\AnsTranslator\\results\\package4.anf')`

ImportANFv2

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

UI Access	**File** > **Import** > **ANF**.		
Parameters	Name	Type	Description
	<ANFfilename>	String	Full path of ANF file.
	<outputPathName>	String	Full path of *.aedb output file.
	<controlFileName>	String	Full path of XML control file.

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

	<*cmpFileName*>	String	Full path of CMP file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportANFv2 (<*ANFfilename*>, <*outputPathName*>, <*controlFileName*>, <*cmpFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportANFV2("C:/Users/jdoe/Documents/SAMPLEFILES/my_model.anf",`
	`"C:/Users/jdoe/Documents/Ansoft/my_model.aedb",`
	`"C:/Users/jdoe/Documents/Ansoft/my_model.xml", "")`

ImportAutoCAD

Imports an AutoCAD file into a new project.

UI Access	**File** > **Import** > **AutoCAD**.		
Parameters	Name	Type	Description
	<*dxfFileName*>	String	Full path of DXF file.
	<*outputPathFileName*>	String	Full path of EDB file to create during import.
	<*controlFileName*>	String	Full path of XML control file. Pass empty string if 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	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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*xmlFileName*></td><td>String</td><td>Full path of XML file.</td></tr><tr><td><*outputPathName*></td><td>String</td><td>Full path of output file.</td></tr><tr><td><*logFileName*></td><td>String</td><td>Full path of log file.</td></tr></table>
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')`

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

ImportEDB

Imports an EDB file into a new project.

UI Access	**File** > **Import** > **EDB**.
Parameters	Name | Type | Description *<edbFileName>* | String | Full path of EDB file.
Return Value	None.

Python Syntax	ImportEDB (*<edbFileName>*)

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

ImportExtracta

Imports a Cadence Extracta file into a new project.

> **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**.

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><*extractaFileName*></td><td>String</td><td>Full path of Extracta file.</td></tr><tr><td><*outputPathName*></td><td>String</td><td>Full path of output file to be created.</td></tr><tr><td><*controlFileName*></td><td>String</td><td>Optional. Full path to XML control file.</td></tr></table>
Return Value	None.

Python Syntax	ImportExtracta (<*extractaFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportExtracta('C:/MyFiles/projectforimport.brd', 'C:/MyFiles/project.edb', '')`

ImportGDSII

Imports a GDSII file into a new project.

UI Access	**File** > **Import** > **GDSII**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></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></table>

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

	<mapFileName>	String	If technology file was used in place of the control file. Full path to either gdsii mapping file (corresponding to `-g=gdsMappingFile` argument in anstranslator) or XML control file. Otherwise, pass empty string.
Return Value	None.		

Python Syntax	ImportGDSII(*<gdsiiFileName>*, *<outputPathName>*, *<controlFileName>*, *<mapFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportGDSII('C:/Files/test.gds', 'C:/Files/test.aedb',`
	`'C:/Files/test.ircx', 'C:/Files/test.xml')`

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.		

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

Python Syntax	ImportGerber(<*gerberFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportGERBER('C:/Files/test.gbr', 'C:/Files/test.aedb.edb',`
	`'C:/Files/test.xml')`

ImportIDF

Imports an IDF file into a new project. To merge with an existing design, use ImportIDFandMerge.

UI Access	placeholder

Parameters

Name	Type	Description
<*NAME*>	string	Settings
<*Board*>	bool	Full path of board file
<*Library*>	int	Full path of library file
<*Control*>	int	Full path of control file
<*Filters*>	list	["Cap","Height","HeightExclude2D","Ind","Power","Res"]
<*CreateFilteredAsNonModel*>	bool	True or False
<*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

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

	<NumInternalLayer>	int	Number of internal layers
	<HighSurfaceCopper>	int	High side surface coverage percentage
	<LowSurfaceCopper>	int	Low side surface coverage percentage
	<InternalLayerCopper>	int	Internal layer coverage percentage
	<TraceMaterial>	string	Trace material name
	<SubstrateMaterial>	int	Substrate material name
	<CreateBoard>	bool	True or False
	<ModelBoardAsRect>	bool	True or False
	<ModelDeviceAsRect>	bool	True or False
	<Cutoff>	bool	True or False
	<IncludeDrilledHoles>	bool	True or False
	<HoleDiameterCutoff>	string	Hole diameter size and unit
	<ReplaceDevices>	bool	True or False
	<CreatePointsOnBoardForInstances>	bool	True or False
Return Value	None.		

Python Syntax	ImportIDF_1 (*<NAME>*, *<Board>*, *<Library>*, *<Control>*, *<Filters>*, *<CreateFilteredAsNonModel>*, *<NAME>*, *<NAME>*, *<HighSurfThickness>*, *<LowSurfThickness>*, *<InternalLayerThickness>*, *<NumInternalLayer>*, *<HighSurfaceCopper>*, *<LowSurfaceCopper>*, *<InternalLayerCopper>*, *<TraceMaterial>*, *<SubstrateMaterial>*, *<CreateBoard>*, *<ModelBoardAsRect>*, *<ModelDeviceAsRect>*, *<Cutoff>*, *<IncludeDrilledHoles>*, *<ReplaceDevices>*)

Python Example	`oDesign.ImportIDF(`
	`  [`
	`  "NAME:Settings",`
	`  "Board:=" , "C:\\Users\\Model Files\\brd_board.emn",`
	`  "Library:=" , "C:\\Users\\Model Files\\brd_board.emp",`

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

```
                "Control:="             , "C:\\Users\\Model Files\\brd_board.xml",
                "Filters:="             , ["HeightExclude2D"],
                "CreateFilteredAsNonModel:=", False,
                "FootPrint:="           , "0.1mm2",
                [
                        "NAME:definitionOverridesMap"
                ],
                [
                        "NAME:instanceOverridesMap"
                ],
                "HighSurfThickness:="   , "0.07mm",
                "LowSurfThickness:="    , "0.07mm",
                "InternalLayerThickness:=", "0.07mm",
                "NumInternalLayer:="    , 2,
                "HighSurfaceCopper:="   , 30,
                "LowSurfaceCopper:="    , 30,
                "InternalLayerCopper:=" , 30,
                "TraceMaterial:="       , "Cu-Pure",
                "SubstrateMaterial:="   , "FR-4",
                "CreateBoard:="         , True,
                "ModelBoardAsRect:="    , True,
                "ModelDeviceAsRect:="   , False,
                "Cutoff:="              , False,
                "IncludeDrilledHoles:=" , True,
                "HoleDiameterCutoff:="  , "0.1mm",
                "ReplaceDevices:="      , False,
                "CreatePointsOnBoardForInstances:=", False
        ])
```


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

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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*idfFileName*></td><td>String</td><td>Full path of GERBER file.</td></tr><tr><td><*libFileName*></td><td>String</td><td>Optional. Full path of library file. Pass empty string if none.</td></tr><tr><td><*controlFileName*></td><td>String</td><td>Optional. Full path of XML control file. Pass empty string if none.</td></tr><tr><td><*project*></td><td>String</td><td>Name of project to merge with.</td></tr><tr><td><*design*></td><td>String</td><td>Name of design to merge with.</td></tr></table>
Return Value	None.

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

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

ImportIDX

Imports and IDX model into a new Icepak project.

UI Access	**Icepak** > **Import IDX**

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 .idx file
<Library>	string	NA
<Control>	string	NA
<Filters>	list	HeightExclude2D
<CreateFilteredAsNonModel>	bool	True or False
<NAME>	string	definitionOverridesMap
<NAME>	string	instanceOverridesMap
<HighSurfThickness>	string	High surface layer thickness value and unit
<LowSurfThickness>	string	Low surface layer thickness value and unit
<InternalLayerThickness>	string	Internal surface layer thickness value and unit
<NumInternalLayer>	int	Number of internal layers value
<HighSurfaceCopper>	int	High surface percent coverage value
<LowSurfaceCopper>	int	Low surface percent coverage value
<InternalLayerCopper>	int	Internal layer percent coverage value
<TraceMaterial>	string	Trace material name
<SubstrateMaterial>	int	Substrate material name
<CreateBoard>	bool	True or False
<ModelBoardAsRect>	bool	True or False
<ModelDeviceAsRect>	bool	True or False
<Cutoff>	bool	True or False
<ReplaceDevices>	bool	True or False

Return Value

None

Python Syntax

ImportIDX (*<NAME>*, *<Board>*, *<Library>*, *<Control>*, *<Filters>*, *<CreateFilteredAsNonModel>*, *<NAME>*, *<NAME>*, *<HighSurfThickness>*, *<LowSurfThickness>*, *<InternalLayerThickness>*, *<NumInternalLayer>*,

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

	<*HighSurfaceCopper*>, <*LowSurfaceCopper*>, <*InternalLayerCopper*>, <*TraceMaterial*>, <*SubstrateMaterial*>, <*CreateBoard*>, <*ModelBoardAsRect*>, <*ModelDeviceAsRect*>, <*Cutoff*>, <*ReplaceDevices*>)

Python Example	`oDesign.ImportIDX(`
	`[`
	`"NAME:Settings",`
	`"Board:=" , "C:\\Users\\Model files\\PCB-00278_A.idx",`
	`"Library:=" , "",`
	`"Control:=" , "",`
	`"Filters:=" , ["HeightExclude2D"],`
	`"CreateFilteredAsNonModel:=", False,`
	`[`
	`"NAME:definitionOverridesMap"`
	`],`
	`[`
	`"NAME:instanceOverridesMap"`
	`],`
	`"HighSurfThickness:=" , "0.07mm",`
	`"LowSurfThickness:=" , "0.07mm",`
	`"InternalLayerThickness:=", "0.07mm",`
	`"NumInternalLayer:=" , 2,`
	`"HighSurfaceCopper:=" , 30,`
	`"LowSurfaceCopper:=" , 30,`
	`"InternalLayerCopper:=" , 30,`
	`"TraceMaterial:=" , "Cu-Pure",`
	`"SubstrateMaterial:=" , "FR-4",`
	`"CreateBoard:=" , True,`
	`"ModelBoardAsRect:=" , False,`
	`"ModelDeviceAsRect:=" , False,`
	`"Cutoff:=" , False,`
	`"ReplaceDevices:=" , False`
	`])`

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

ImportIPC

Imports an IPC2581 file into a new project.

UI Access	**File** > **Import** > **IPC2581**.		
Parameters	Name	Type	Description
	<ipcFileName>	String	Full path of IPC2581 file.
	<outputPathName>	String	Full path of EDB file to create during import.
	<controlFileName>	String	Optional. Full path of XML control file. Pass empty string if none.
Return Value	None.		

Python Syntax	ImportIPC(*<ipcFileName>*, *<outputPathName>*, *<controlFileName>*)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportIPC('C:/Files/test.cvg', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml', 'C:/Files/test.txt')`

ImportODB

Imports an ODB++ file into a new project.

UI Access	**File** > **Import** > **ODB++**.

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><*odbFileName*></td><td>String</td><td>Full path of ODB++ 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. Pass empty string if none.</td></tr></table>
Return Value	None.

Python Syntax	ImportODB(<*odbFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportODB('C:/Files/test.odb', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml')`

ImportXFL

Imports an XFL file into a new project.

UI Access	**File** > **Import** > **XFL**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*xflFileName*></td><td>String</td><td>Full path of XFL 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. Pass empty string if none.</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	ImportXFL(<*xflFileName*>, <*outputPathName*>, <*controlFileName*>)

Python Example	`oDesktop.RestoreWindow()`
	`oTool = oDesktop.GetTool('ImportExport')`
	`oTool.ImportXFL('C:/Files/test.xfl', 'C:/Files/test.aedb',`
	`'C:/Files/test.xml')`

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

5 - Running Instances Manager Script Commands

The Running Instances Manager is a scripting object that lets you identify and connect to all running instances of Electronics Desktop. oDesktop objects that are returned provide full scripting functionality. Running Instances Manager commands should be executed by the oDesktop object. For example:

```
oRunningInstances = oDesktop.GetRunningInstancesMgr()
```

GetAllRunningInstances

GetRunningInstanceByProcessID

GetRunningInstanceByProject

GetAllRunningInstances

Returns a list of running instances of Ansys Electronics Desktop.

UI Access	N/A
Parameters	None.
Return Value	Array containing list of Ansys Electronics Desktop instances.

Python Syntax	GetAllRunningInstances()
Python Example	`obj = oRunningInstances.GetAllRunningInstances()`

GetRunningInstanceByProcessID

Returns the instance of Ansys Electronics Desktop that is running a specified process.

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

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><processID></td><td>Integer</td><td>Process ID</td></tr></table>
Return Value	String of the returned instance.

Python Syntax	GetRunningInstanceByProcessID(<processID>)
Python Example	`obj = oRunningInstances.GetRunningInstanceByProcessID(12345)`

GetRunningInstancesMgr

Returns the object of the Running Instances Manager.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>None</td><td></td><td></td></tr></table>
Return Value	Running instances manager object

Python Syntax	GetRunningInstancesMgr()
Python Example	`oRunningInstances = oDesktop.GetRunningInstancecMgr()`

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

6 - Project Object Script Commands

Project commands should be executed by the oProject object.

One example of accessing this object is:

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

Dataset Script Commands:

AddDataset

DeleteDataset

EditDataset

ExportDataset

HasDataset

ImportDataset

Other Project Object Script Commands:

AddMaterial

AnalyzeAll

ChangeProperty

ClearMessages

CloneMaterial

Close

CopyDesign

CutDesign

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

DeleteDesign

DeleteToolObject

EditMaterial

ExportMaterial

GetActiveDesign

GetArrayVariables

GetChildNames [Project]

GetChildObject [Project]

GetChildTypes [Project]

GetDefinitionManager

GetDependentFiles

GetDesign

GetDesigns

GetEDBHandle

GetLegacyName

GetName [Project]

GetObjPath [Project]

GetPath

GetPropEvaluatedValue

GetPropNames [Project]

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

GetPropSIValue

GetPropValue [Project]

GetProperties

GetPropertyValue

GetTopDesignList

GetVariableValue

GetVariables

InsertDesign

InsertDesign (Layout Editor)

InsertDesignWithWorkflow

InsertToolObject

Paste [Project Object]

Redo [Project Level]

RemoveAllUnusedDefinitions

RemoveMaterial

RemoveUnusedDefinitions

Rename

RunToolkit

Save

SaveAs

SaveAsStandAloneProject

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

SaveProjectArchive

SetActiveDefinitionEditor

SetActiveDesign

SetPropValue [Project]

SetPropertyValue

SetVariableValue

SimulateAll

Undo [Project]

UpdateDefinitions

AddDataset

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

UI Access	**Project** > **Datasets** > **Add**		
Parameters	Name	Type	Description
	<*DatasetDataArray*>	Array	["NAME:<DatasetName>", ["NAME:Coordinates", <CoordinateArray>, <CoordinateArray>, ...]
	<*DatasetName*>	String	Name of the dataset.
	<*CoordinateArray*>	Array	["NAME:Coordinate", "X:=", <double>, "Y:=",<double>]
Return Value	None.		

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

Python Syntax	AddDataset <*DatasetDataArray*>

Python Example	`oProject.AddDataset(`
	`["NAME:$ds1",`
	`  ["NAME:Coordinates",`
	`  ["NAME:Coordinate",`
	`  "X:=", 2,`
	`  "Y:=", 4`
	`  ],`
	`  ["NAME:Coordinate",`
	`  "X:=", 6,`
	`  "Y:=", 8`
	`  ]`
	`  ]`
	`]`
	`)`

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>, ...]

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

<*MatProperty*>	Array	For simple material: "<PropertyName>:=", <value> For anisotropic material: ["NAME:<PropertyName>", "property_type:=", "AnisoProperty", "unit:=", <Unit>", "component1:=", <value>, "component2:=", <value>, "component3:=", <value>))]
<*PropertyName*>	String	Should be one of the following (depending on the material, design, and solution types): 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"

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

		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" 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"

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	AddMaterial (["NAME:<MaterialName>", <MatProperty>, <MatProperty>, ...])

Python Example	`oDefinitionManager.AddMaterial(`
	`["permittivity:=", "2.2", "0.002"])`
	
	`oDefinitionManager.AddMaterial(`
	`["NAME:Material2",`
	`  "dielectric_loss_tangent:=", "44",`
	`  ["NAME:saturation_mag",`
	`  "property_type:=", "AnisoProperty",`
	`  "unit:=", "Gauss",`
	`  "component1:=", "11",`
	`  "component2:=", "22",`
	`  "component3:=", "33"],`
	`  "delta_H:=", "44Oe"])`

AnalyzeAll [project]

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

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()`

ChangeProperty

Changes the properties of an object in the history tree.

UI Access	Right-click an object in the History Tree and select **Properties**.
Parameters	Name: *<propertyArgs>* — Type: Array — Description: Structured array. The properties vary depending on the object. Due to the number of potential configurations, it is recommended that you generate this script using the UI's **Automation** tab.
Return Value	None.

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

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	**Example: Changing the Position of a Box**
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DCmdTab",`
	`  ["NAME:PropServers" , "Box1:CreateBox:1" ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Position", "X:=" , "0.35in", "Y:=" , "0.55in", "Z:=" , "0in"`
	`  ]`
	`  ]`
	`  ]`
	`  ])`
	
	Example: Offset the Position of a 3D Component or Layout Component with a Variable
	`oDesign.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:LocalVariableTab",`
	`  ["NAME:PropServers",`
	`  "LocalVariables"`
	`  ],`
	`  ["NAME:NewProps",`
	`  ["NAME:zH",`
	`  "PropType:=" , "VariableProp",`

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

```
          "UserDef:=" , True,
          "Value:=" , "50mm"
          ]
        ]
      ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers",
        "LC1_1"
      ],
      ["NAME:ChangedProps",
        ["NAME:Position",
        "X:=" , "0mm",
        "Y:=" , "0mm",
        "Z:=" , "zH"
        ]
      ]
    ]
  ])
```

Example: Changing a Box's Material and Wireframe Display

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers" , "Box1" ],
```


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

```
        ["NAME:ChangedProps",
        ["NAME:Material", "Value:=" , "\"vacuum\"" ],
        ["NAME:Display Wireframe", "Value:=" , True ]
      ]
    ]
  ])
```

Example: Change Default Handle (reference coordinate system) for a 3D Component or Layout Component

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers","LC1_1"],
      ["NAME:ChangedProps",
        ["NAME:Handle","Value:=" , " NotchOutY"]
      ]
    ]
  ])

oEditor.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:Maxwell2D",
       ["NAME:PropServers", "Design Settings" ],
      ["NAME:ChangedProps",
        ["NAME:Model settings/Depth", "Value:=" , "2.4mm" ]
       ]
```


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

	]])

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()`

CloneMaterial

Clones a local material.

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><*matName*></td><td>String</td><td>Name of existing material.</td></tr><tr><td><*newName*></td><td>String</td><td>Name for newly cloned material.</td></tr></table>
Return Value	Boolean: • 1 - Material is cloned. • 0 - Existing material not found or a conflict with the new material name.

Python Syntax	CloneMaterial (<*matName*>, <*newName*>)
Python Example	`oDefinitionManager.CloneMaterial("copper1", "copper3")`

Close

Closes the active project.

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

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	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")`

CutDesign

Cuts a design from the active project. The design is stored in memory and can be pasted.

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

> **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 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')`

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("CircuitDesign2")`

DeleteToolObject

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

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 <*ObjectName*> | String | Name of the tool object.
Return Value	None.

Python Syntax	DeleteToolObject(<*ObjectName*>)
Python Example	`oProject.DeleteToolObject("object1")`

EditDataset

Modifies a dataset. This can be executed by the oProject variables.

UI Access	**Project** > **Datasets** > **Edit**.
Parameters	Name | Type | Description <*OriginalName*> | String | Name of the original dataset. <*DatasetDataArray*> | Array | Data for the modified dataset.
Return Value	None.

Python Syntax	EditDataset (<*OriginalName*> <*DatasetDataArray*>)
Python Example	`oProject.EditDataset ("ds1"`

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

```
["NAME:ds2",
  ["NAME:Coordinates",
    [
      "NAME:Coordinate",
      "X:=", 1, "Y:=", 2
    ],
    [
      "NAME:Coordinate",
      "X:=", 3, "Y:=", 4
    ]
  ]
]
)
```

EditMaterial

Modifies an existing material.

UI Access	**View/Edit Materials** command in the material editor

Parameters

Name	Type	Description
<*OriginalName*>	String	Name of the material before editing.
<*MatProperties*>	Array	Structured array containing material properties: [

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

		"NAME:<New material name>", "CoordinateSystemType:=", <string>, "BulkOrSurfaceType:=" , <integer>, ["NAME:PhysicsTypes", "set:=" , <array containing string physics types>], <Optional ModifierDataArray>, "permeability:=" , <string containing value>, "conductivity:=" , <string containing value>, "thermal_conductivity:=", <string containing value>, "mass_density:=" , <string containing value>, "specific_heat:=" , <string containing value>, "youngs_modulus:=" , <string containing value>, "poissons_ratio:=" , <string containing value>, "thermal_expansion_coefficient:=", <string containing value>]
<ModifierDataArray>	Array	Optional structured array containing thermal or spatial modifiers: ["NAME:ModifierData",

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

```
  [
    "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">,
        "use_free_form:=" , <Boolean>,
        "free_form_value:=" , <string modifier value>,
      ]
    ]
  ]
]
```

Return Value	None.

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

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

Python Example	**Without Modifiers:**
	`oDefinitionManager.EditMaterial("alumina_92pct",`
	`[`
	`  "NAME:alumina_92pct",`
	`  "CoordinateSystemType:=" , "Cartesian",`
	`  "BulkOrSurfaceType:=" , 1,`
	`  [`
	`  "NAME:PhysicsTypes",`
	`  "set:=" , ["Electromagnetic","Thermal","Structural"]`
	`  ],`
	`  "permittivity:=" , "9.3",`
	`  "dielectric_loss_tangent:=" , "0.008",`
	`  "thermal_conductivity:=" , "26",`
	`  "mass_density:=" , "3720",`
	`  "specific_heat:=" , "790",`
	`  "youngs_modulus:=" , "267000000000",`
	`  "poissons_ratio:=" , "0.26",`
	`  "thermal_expansion_coeffcient:=", "7.2e-006"`

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

```
]
)
```

With Thermal Modifier:

```
oDefinitionManager.EditMaterial("copper",
  [
  "NAME:copper",
  "CoordinateSystemType:=", "Cartesian",
  "BulkOrSurfaceType:=" , 1,
    [
    "NAME:PhysicsTypes",
    "set:=" , ["Electromagnetic","Thermal","Structural"]
    ],
  [
  "NAME:ModifierData",
    [
    "NAME:ThermalModifierData",
    "modifier_data:=" , "thermal_modifier_data",
      [
      "NAME:all_thermal_modifiers",
        [
        "NAME:one_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.

```
        "Property::=" , "permittivity",
        "Index::=" , 0,
        "prop_modifier:=" , "thermal_modifier",
        "use_free_form:=" , True,
        "free_form_value:=" , "if(Temp > 1000cel, 1, if(Temp < -273.15cel, 1, 1))"
        ]
      ]
    ]
  ],
  "permeability:=" , "0.999991",
  "conductivity:=" , "58000000",
  "thermal_conductivity:=" , "400",
  "mass_density:=" , "8933",
  "specific_heat:=" , "385",
  "youngs_modulus:=" , "120000000000",
  "poissons_ratio:=" , "0.38",
  "thermal_expansion_coefficient:=" , "1.77e-05"
])
```

Transient Solve, Non-linear Drude Data Plasma

```
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("Drude_plasma_parameters_r231")
oDefinitionManager = oProject.GetDefinitionManager()
oDefinitionManager.EditMaterial("Drude",
  [
    "NAME:Drude",
    "CoordinateSystemType:=", "Cartesian",
    "BulkOrSurfaceType:="   , 1,
    [
      "NAME:PhysicsTypes",
      "set:="                , ["Electromagnetic"]
    ],
    [
      "NAME:AttachedData",
      [
        "NAME:MatNonLinearDrudeFreqDepData",
        "property_data:="     , "nonlinear_drude_data",
        "EpsilonInfinity:="   , "1",
        "PlasmaFrequency:="   , "4.62348462366278GHz",
        "CollisionFrequency:=" , "0.00054491190162662GHz",
```


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

```
            "FieldBreakdown:="     , "10000V_per_meter",
            "PlasmaMaintainFrequency:=", "2.31174231183139GHz",
            "NeutralDensity:="     , 2.65164580488373E+20,
            "ElectronDensity:="    , 2.65164580488373E+17,
            "CollisionRateConstant:=", 2.05499505485618E-15
        ]
    ]
])
```

ExportDataset

Exports a dataset to a named file. This can be executed by the oProject variables.

UI Access	**Project** > **Datasets** > **Export**.
Parameters	Name | Type | Description *<datasetFileFullPath>* | String | The full path to the file.
Return Value	None.

Python Syntax	ExportDataset (<*datasetFileFullPath*>)
Python Example	`oProject.ExportDataset('e:/tmp/dsdata.txt')`

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

ExportMaterial

Exports a local material to a library.

UI Access	**Export to Library** command in the material editor.
Parameters	See table below
Return Value	None.

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.

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 | Description *<type>* | String | (Optional) Default is "design"; Any value returned by GetChildTypes() can be used.
Return Value	Array of names of children for the queried object.

Python Syntax	GetChildNames (*<type>*)
Python Example	`arrDesignNames = oProject.GetChildNames()` `arrVarbleNames = oProject.GetChildNames("Variable")`

GetChildObject [Project]

Returns the project's child objects.

> **Note:**
>
> `GetChildObject` will return normally if there are no active objects.

UI Access	N/A
Parameters	Name | Type | Description *<path>* | String | The path may include multiple generations (for example, "designOb-

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

	ject/moduleObj/SetupObject"). See Object Path.
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.
Return Value	List of referenced files.

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

Python Syntax	GetDependentFiles()
Python Example	`files = oProject.GetDependentFiles()`

GetDesign

Returns the interface to a specific design in a given project.

UI Access	N/A
Parameters	Name | Type | Description *<designName>* | String | Name of the design.
Return Value	Object of the specified design.

Python Syntax	GetDesign (*<designName>*)
Python Example	`oProject.GetDesign("HFSSDesign1")`

GetDesigns

Obtains all designs in the current project.

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

UI Access	N/A
Parameters	None.
Return Value	List of objects for all designs in the project.

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

GetEDBHandle

Returns the EDB handle for the project.

> **Important:**
>
> This script is for internal Ansys use only.

UI Access	N/A
Parameters	None.
Return Value	String indicating the EDB handle for the project.

Python Syntax	GetEDBHandle()

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.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

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 containing the project name, not including the path or extension.

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

GetObjPath [Project]

Obtains the project name from the full path.

UI Access	N/A
Parameters	None.
Return Value	String containing only the project name.

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

GetPath

Returns the location of the project on disk.

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

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the project, not including the project name.

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

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A		
Parameters	Name	Type	Description
	<PropName>	String	Name of the property.
Return Value	String value of the evaluated value.		

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

Python Syntax	GetPropEvaluatedValue (<*PropName*>)
Python Example	`oVar = oDesign.GetChildObject(" Variables/var")` `oVar.GetPropEvaluatedValue()`

GetPropNames [Project]

Obtains the property name of the object. At the project level, GetPropNames always returns empty because the project is not associated with any property.

UI Access	N/A
Parameters	Name: *<includeReadOnly>* Type: Boolean Description: (Optional) • True – Include read only props. • False – Do not include read only props.
Return Value	Empty array.

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

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

GetPropSIValue

Returns the SI-Value for Value-Property and Variable. Return NAN for other property type if its value is not able to convert to be a double-floating point value.

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PropName*></td><td>String</td><td>Name of the property.</td></tr></table>
Return Value	Property value as a double floating value, or NAN if the property value cannot be converted to double floating point.

Python Syntax	GetPropSIValue (<*PropName*>)

Python Example	`oCreateBox = oDesign.GetChildObject("3D Modeler/Box1/CreateBox:1")`
	`oCreateBox.GetPropValue("xSize")`
	return "length / 2"
	`oCreateBox.GetPropEvaluatedValue("xSize")`
	return '0.4mm'
	`oCreateBox.GetPropSIValue("xSize")`

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

	return 0.0004

GetPropValue [Project]

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

UI Access	N/A
Parameters	Name | Type | Description <*propPath*> | String | A child object's property path. See: Property Function Summary.
Return Value	String of property value.

Python Syntax	GetPropValue(<*propPath*>)
Python Example	`oProject.GetPropValue("TeeModel/offset")`

GetProperties

Gets a list of all the properties belonging to a specific <PropServer> and <PropTab>. This can be executed by the oProject, oDesign, or oEditor variables.

UI Access	N/A		
Parameters	Name	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")

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

			• ProjectVariableTab ("Project variables") • ConstantsTab ("Constants") • BaseElementTab ("Symbol" or "Footprint") • ComponentTab ("General") • Component("Component") • CustomTab ("Intrinsic Variables") • Quantities ("Quantities") • Signals ("Signals")
	<PropServer>	String	An object identifier, generally returned from another script method, such as 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>

<table>
<tr><td>Python Syntax</td><td>GetPropertyValue (<PropTab>, <PropServer>, <PropName>)</td></tr>
<tr><td>Python Example</td><td>

```
selectionArray = oEditor.GetSelections()
for k in selectionArray:
val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")
...
```

</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.

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: String — Description: Name of the variable to access. (<*VarName*>)

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.

ImportDataset

Imports a dataset from a named file. This can be executed by the oProject variables. The name of the dataset is filename+index number (e.g., dsdata1) unless the filename ends with a trailing number. When there is a trailing number at the end, we will remove the number and use first unused index. Alternatively, the name of the dataset can be explicitly defined by providing a string as an optional second argument.

UI Access	**Project** > **Datasets** > **Import**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*datasetFileFullPath*></td><td>String</td><td>The full path to the file containing the dataset values. *.tab files recommended (see note below).</td></tr><tr><td><*optionalDatasetName*></td><td>String</td><td>*Optional*. User-defined dataset name.</td></tr></table>
Return Value	None.

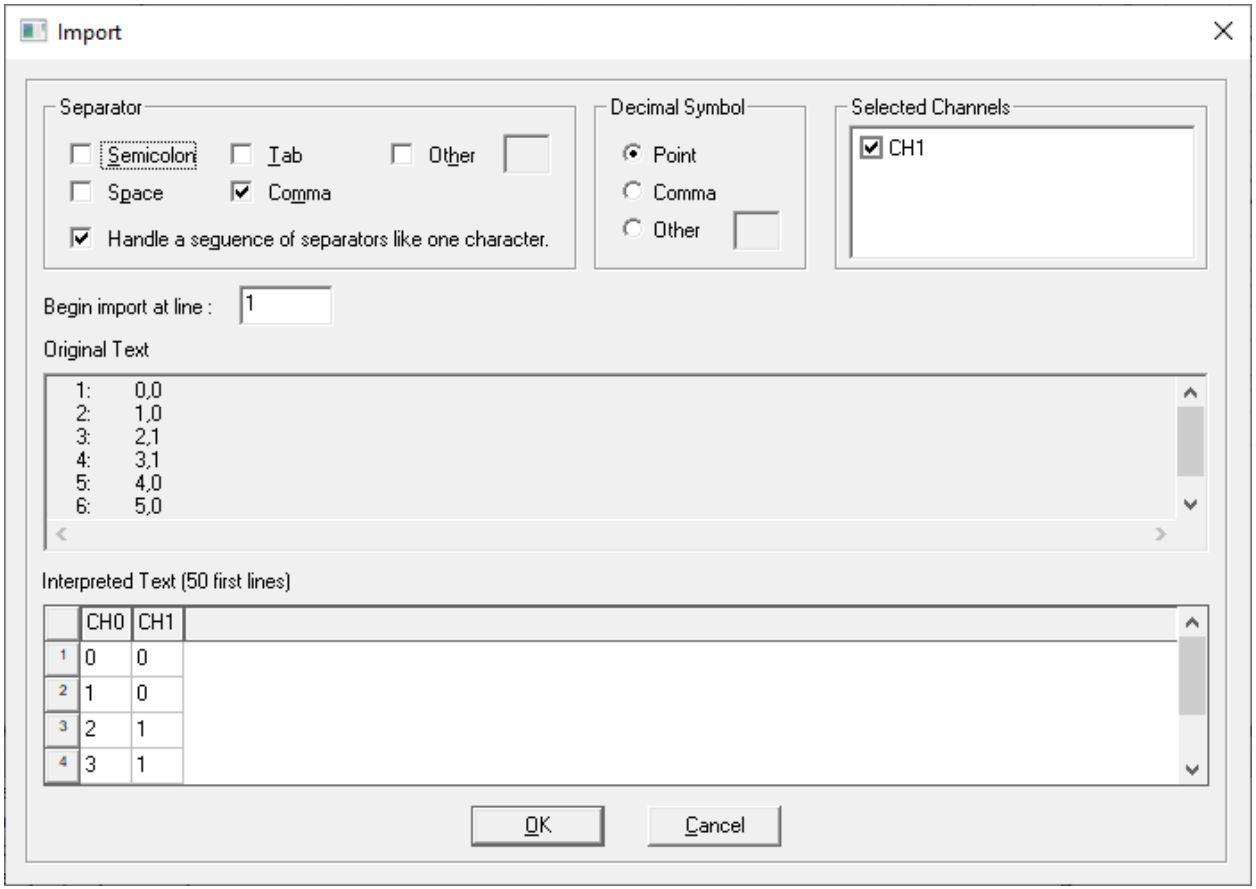
Python Syntax	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')` `oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

Note About File Types:

Tab-delimited or space-delimited files with the extension *.tab are the recommended file type. When using ImportDataset at the Design level, *.tab is the only file type supported.

At the Project level, other file types are supported (for example, *.csv). However, after calling the command, you must configure the file import format manually through the Electronics Desktop GUI by selecting **Project** > **Datasets** and clicking **Import**.

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



InsertDesign

Inserts a new design in the project. For Circuit and HFSS 3D Layout scripts, the last argument will always be empty.

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

UI Access	**Project** > **[Insert Circuit Design / Insert Circuit Netlist / Insert HFSS 3D Layout Design]**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*DesignType*></td><td>String</td><td>Design type. Valid types: "Circuit", "Nexxim Circuit", "System" , "Nexxim Net-list", "HFSS3D".</td></tr><tr><td><*DesignName*></td><td>String</td><td>Design name.</td></tr><tr><td><*TechnologyFile*></td><td>String</td><td>The path to the Circuit technology file to be used in the design. Use a pair of empty double quotes ("") for none.</td></tr><tr><td><*SubcircuitID*></td><td>String</td><td>(Optional) Must be preceded by a colon if included along with the Technology File name. No colon is necessary when the subcircuit ID is omitted.</td></tr><tr><td>""</td><td>None</td><td>Empty argument.</td></tr></table>
Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*TechnologyFile*>:<*SubCircuitID*>,'')
Python Example	`oProject.InsertDesign("Nexxim Circuit","MyDesign", "C:\\Program Files\\ANSYS Inc\\v261\\AnsysEM", "")`

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></table>

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

	<workflowName>	String	Name of the workflow.
	<specName>	String	Name of the spec.
	<fileName>	String	Name of the file.
	<libLoc>	String	Type of library, such as SysLib.
	<stationaryPath>	String	Path.
Return Value	IDispatch string, such as 'IDispatch(IAltraSimScript)'		

Python Syntax	InsertDesignWithWorkflow(*<type>*, *<workflowName>*, *<specName>*, *<fileName>*, *<libLoc>*, *<stationaryPath>*)
Python 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()`

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

Paste (Project Object)

Pastes a design in the active project.

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()`

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

RemoveAllUnusedDefinitions

Removes all unused project definitions.

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	
Return Value	None.

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".

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

Python Syntax	RemoveMaterial (<*MaterialName*>, <*IsProjectMaterial*>, <*LibraryName*>, <*LibraryLocation*>)
Python Example	`oDefinitionManager.RemoveMaterial ([` `"Material1", false, "mo0907","UserLib"])`

RemoveUnusedDefinitions

Removes any unused project definitions.

UI Access	**Tools** > **Project Tools** > **Remove Unused Definitions**.
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"`

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

```
    ],
    [
      "NAME:SurfaceMaterials",
      "Steel-oxidised-surface"
    ]
  ])
```

Rename

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

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

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

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

RunToolkit

Runs a Python toolkit script, applying it to the active design. The toolkit script itself may have prerequisites, such as sweeps defined, or specific model characteristics, such as port definitions.

> **Important:**
>
> Full scripting for cable modeling is not supported and no arguments are allowed in the script. The **RunToolkit** command will not automatically create a cable bundle, but will simply open the **Cable Modeling** dialog box. You will have to manually input your parameters.

UI Access	[**Circuit**] > **Toolkit** > [**Script Name**].
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*LibraryType*></td><td>String</td><td>Name of the library in which the script is located.</td></tr><tr><td><*ToolkitName*></td><td>String</td><td>Name of the Python script.</td></tr><tr><td><*ToolkitArg*></td><td>Array</td><td>Structured array containing arguments required by the toolkit script.</td></tr></table>
Return Value	User-defined solution(s) and/or output(s).

Python Syntax	RunToolkit(<*LibraryType*>, <*ToolkitName*>, <*ToolkitArg*>)
Python Example	`oDesign.RunToolkit("SysLib", "Hearing Aid Compliance", [])`

Save

Saves the active project.

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_absolute, or empty string.
<*OverwriteActions*>	Array	For Schematic/Layout projects only. Otherwise omit. 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.

			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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*EditorName*></td><td>String</td><td>Name of the definition editor to set active, one of "SymbolEditor", "FootprintEditor".</td></tr><tr><td><*DefinitionName*></td><td>String</td><td>The combination name for the symbol or footprint, <*libname*>:<*def-name*></td></tr></table>
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	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*DesignName*></td><td>String</td><td>Name of the design to set as the active design.</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	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	<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.</td></tr><tr><td><*newValue*></td><td>String</td><td>New property value.</td></tr></table>
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><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 CompInst@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")`

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.

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

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

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.

7 - Design Object Script Commands

Design object commands should be executed by the oDesign object.

```
oDesign.CommandName <args>
```

For example:

```
oDesign = oProject.SetActiveDesign("CircuitDesign1")
```

Conventions Used in this Chapter

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

- Analysis Module – "AnalysisSetup"
- Data Module – “DataBlock”
- Optimetrics Module – "Optimetrics"
- Output Variable Module – "OutputVariable"
- Reporter Module – "ReportSetup"
- User Defined Documents Module – "UserDefinedDocuments"
- User Defined Solutions Module – "UserDefinedSolutionModule"

AddDataset

AddDynamicLink

AddModelingProperties

AnalyzeAll

AnalyzeAllNominal

ApplyMeshOps

ChangePortProperty [Nexxim]

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

ChangeProperty

ChangeSourceProperty [Nexxim]

CopyEyeItemAsCommand

CopyItemCommand

CreateReport

DeleteDataset

DeleteDesignInstance

DeleteSolutionVariation

EditDataset

EditImportData

EditNotes

EditOutputVariable

ExportForHSpice

ExportForSpice

ExportNMFData

ExportLayoutViaCurrentDensity

ExportNetlist

ExportNetworkData

ExportReport

GetAllPorts

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

GetChildNames [Design]

GetChildObject [Design]

GetChildTypes [Design]

GetDesignID

GetDesignType

GetDesignName

GetEditor

GetModule

GetName

GetNoteText

GetObjPath

GetProject

GetPropEvaluatedValue

GetPropHost

GetPropNames [Design]

GetPropSIValue

GetPropValue [Design]

GetProperties

GetPropertyValue

GetResultsDirectory [Nexxim]

GetSourceData (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.

GetVariableValue

GetVariables

GetVariationVariableValue

ImportArray

ImportData [Nexxim]

ImportDataFilePath [Nexxim]

ImportDataset

InsertDesign

PasteDesign

PasteItemCommand

Redo

RemoveImportData (Layout Editor)

RemoveModelingProperties

RenameDesignInstance

RenameImportData [Nexxim]

RenameSource [Nexxim]

ReportTemplates

RunToolkit

SetActiveEditor

SetDesignSettings

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

SetPropValue [Design]

SetPropertyValue

SetShowLayoutForLayoutComponent

SetVariableValue

SimulateLink

Undo

UseCircuitSParameterDefinition

AddDataset

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

UI Access	**Project** > **Datasets** > **Add**
Parameters	<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>
Return Value	None.

Python Syntax	AddDataset <*DatasetDataArray*>
Python Example	`oProject.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]]]) ```

AddDynamicLink

Adds a dynamic link component to a Circuit or HFSS 3D Layout design. Circuit designs can accept dynamic links from HFSS, Q3D, 2D Extractor, and SIwave.

> **Note:**
>
> HFSS 3D Layout designs can only accept dynamic links from HFSS.

UI Access	• Open the Circuit or HFSS 3D Layout design to which you want to add the link. On the **Component Libraries** menu, open the **Symbols** tab, and then click **Import Models**. Select the appropriate icon from the panel of icons. The **Import Components** dialog opens. • If the source for a dynamic link is an HFSS design, in the Project Manager drag and drop the design onto a Circuit or HFSS 3D Layout design to create the link.

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
<*designName*>	String	Design source for the dynamic link. Optional, defaults to the first design in the project pointed to by projectPath.
<*projectPath*>	String	Project source for the dynamic link. Optional, defaults to the current active project.
<*linkName*>	String	Name applied to the dynamic link. Optional, defaults to the designName.
<*solutionName*>	String	Solution in source design which will be referenced by the dynamic link. Optional, defaults to the first setup:sweep.
<*transLinePort* >	String	For HFSS designs only, port name for transmission lines. It is created only if it is *not* empty. Optional, defaults to “”. This value is ignored for non-HFSS design types.
<*ReduceMatrix*>	String	For Q3D and 2D extractor designs, reduce matrix which will be referenced by the dynamic link. This is ignored for other design types. Optional, defaults to “Original”.
<*EnableHFSSCableModeling*>	Boolean	For 2D extractor designs only, flag to create cable models. This is ignored for other designs types. Optional, defaults to false.
<*Description*>	String	Description of the dynamic link's source. Optional.

Return Value

None.

Python Syntax

AddDynamicLink (<*designName*>, <*projectPath*>, <*linkName*>, <*solutionName*>, <*transLinePort*>, <*reducedMatrix*>, <*enableCableModeling*>, <*description*>)

Python Example

```
oDesign = oProject.SetActiveDesign("Circuit1")

oDesign.AddDynamicLink("TeeModel","$PersonalLib\\TeeWrongFile.aedt","TeeLink","Setup1 : LastAdaptive","Port1","",False,"desc")
```


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

AddModelingProperties

Use: Add a modeling property to a design

Command: None

Syntax: AddModelingProperties <design>

Return Value: None

Parameters: <design>

Type: string

AddMaxwellLink

Adds a Maxwell dynamic link component to a Circuit design.

UI Access	
Parameters	

Name	Type	Description
<*designName*>	String	Design source for the dynamic link. Optional, defaults to the first design in the project pointed to by projectPath.
<*projectPath*>	String	Project source for the dynamic link. Optional, defaults to the current active project.
<*linkName*>	String	Name applied to the dynamic link. Optional, defaults to the designName.
<*solutionName*>	String	Solution in source design which will be referenced by the dynamic link. Optional, defaults to the first setup:sweep.
<*ReduceMatrix*>	String	Reduce matrix which will be referenced by the dynamic link. Optional, defaults to Original.

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	AddMaxwellLink (<*designName*>, <*projectPath*>, <*linkName*>, <*solutionName*>, <*reducedMatrix*>, <*description*>)
Python Example	`oDesign = oProject.SetActiveDesign("Circuit1")` `oDesign.AddQ3DRLGCLink("02MaxwellDesign","$PersonalLib\\TestDesign02.aedt")`

AddQ3DRLGCLink

Adds a Q3D RLGC dynamic link component to a Circuit design.

UI Access	
Parameters	

Name	Type	Description
<*designName*>	String	Design source for the dynamic link. Optional, defaults to the first design in the project pointed to by projectPath.
<*projectPath*>	String	Project source for the dynamic link. Optional, defaults to the current active project.
<*linkName*>	String	Name applied to the dynamic link. Optional, defaults to the designName.
<*solutionName*>	String	Solution in source design which will be referenced by the dynamic link. Optional, defaults to the first setup:sweep.
<*ReduceMatrix*>	String	Reduce matrix which will be referenced by the dynamic link. Optional, defaults to Original.

Return Value	None.

Python Syntax	AddQ3DRLGCLink (<*designName*>, <*projectPath*>, <*linkName*>, <*solutionName*>, <*reducedMatrix*>, <*descrip-*

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

	tion>)
Python Example	`oDesign = oProject.SetActiveDesign("Circuit1")` `oDesign.AddQ3DRLGCLink("02_Q3DDesign","$PersonalLib\\TestDesign02.aedt")`

AnalyzeAll [design]

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

UI Access	N/A		
Parameters	Name	Type	Description
	isBlocking	Boolean	An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None		

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

AnalyzeAllNominal

Runs all defined 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 **Analysis** in the project tree, and select **Analyze All**.
Parameters	Name: isBlocking; Type: Boolean; Description: An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

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

ChangePortProperty [Nexxim]

Change a port property

UI Access	Right click port > Properties
Parameters	Name: <Port Name>; Type: string; Description: port name Name: <Port Info>; Type: array; Description: Array of attributes Name: <Port Properties>; Type: array; Description: Array of attributes and values
Return Value	None

P-	`ChangePortProperty(<portName> [<portInfo>] [<portProperties>])`

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

y-t-h-o-n S-y-n-t-a-x	
P-y-t-h-o-n E-x-a-m-p-l-e	oDesign.ChangePortProperty("PortName", ["NAME:PortName", "IIPortName:=", "PortName", "SymbolType:=", 0],["NAME:Properties", ["NAME:NewProps", ["NAME:term", "PropType:=", "TextProp", "OverridingDef:=", True, "Value:=", "Model1"]], ["NAME:ChangedProps", ["NAME:TerminationData", "Value:=", "Zo"], ["NAME:pnum", "Value:=", "2"], ["NAME:noisetemp", "Value:=", "16.85"]]])

ChangeProperty

Changes the properties of an object in the history tree.

UI Access	Right-click an object in the History Tree and select **Properties**.
Parameters	

Name	Type	Description
<*propertyArgs*>	Array	Structured array. The properties vary depending on the

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

			object. Due to the number of potential configurations, it is recommended that you generate this script using the UI's **Automation** tab.
Return Value	None.		

Python Syntax	ChangeProperty(<*propertyArgs*>)

Python Example	**Example: Changing the Position of a Box**
	`oEditor.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:Geometry3DCmdTab",`
	`  ["NAME:PropServers" , "Box1:CreateBox:1" ],`
	`  ["NAME:ChangedProps",`
	`  ["NAME:Position", "X:=" , "0.35in", "Y:=" , "0.55in", "Z:=" , "0in"`
	`  ]`
	`  ]`
	`  ]`
	`  ])`
	
	Example: Offset the Position of a 3D Component or Layout Component with a Variable
	`oDesign.ChangeProperty(`
	`  ["NAME:AllTabs",`
	`  ["NAME:LocalVariableTab",`
	` ["NAME:PropServers",

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

```
          "LocalVariables"
        ],
        ["NAME:NewProps",
          ["NAME:zH",
          "PropType:=" , "VariableProp",
          "UserDef:=" , True,
          "Value:=" , "50mm"
          ]
        ]
      ]
    ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.ChangeProperty(
    ["NAME:AllTabs",
      ["NAME:General",
        ["NAME:PropServers",
          "LC1_1"
        ],
        ["NAME:ChangedProps",
          ["NAME:Position",
          "X:=" , "0mm",
          "Y:=" , "0mm",
          "Z:=" , "zH"
          ]
        ]
      ]
    ])
```


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

Example: Changing a Box's Material and Wireframe Display

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:Geometry3DAttributeTab",
      ["NAME:PropServers" , "Box1" ],
       ["NAME:ChangedProps",
       ["NAME:Material", "Value:=" , "\"vacuum\"" ],
       ["NAME:Display Wireframe", "Value:=" , True ]
      ]
    ]
  ])
```

Example: Change Default Handle (reference coordinate system) for a 3D Component or Layout Component

```
oEditor.ChangeProperty(
  ["NAME:AllTabs",
    ["NAME:General",
      ["NAME:PropServers","LC1_1"],
      ["NAME:ChangedProps",
        ["NAME:Handle","Value:=" , " NotchOutY"]
      ]
    ]
  ])

oEditor.ChangeProperty(
  ["NAME:AllTabs",
     ["NAME:Maxwell2D",
```


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", "Design Settings"], ["NAME:ChangedProps", ["NAME:Model settings/Depth", "Value:=" , "2.4mm"]]]]) ```

ChangeSourceProperty [Nexxim]

Change a source property.

UI Access	Right click source > Properties
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><Source_Name></td><td>string</td><td>Source name</td></tr><tr><td><Properties></td><td>array</td><td>Array of source attributes and their values</td></tr></table>
Return Value	None

Python Syntax	ChangeSourceProperty(<sourceName>,[<Properties])
Python Example	``` oDesign.ChangeSourceProperty("SourceName", [["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.

```
                [ "NAME:NewProps",
                        [ "NAME:noise", "PropType:=", "TextProp", "OverridingDef:=", True, "UserDef:=",
"Value:=", "SourceNoise1" ],
                        [ "NAME:mod", "PropType:=", "TextProp", "OverridingDef:=", True, "UserDef:=", T
"Value:=", "SourceModulation1"] ] ],
                [ "NAME:DataBlocks",
                        [ "NAME:NewDataBlk",
                                [ "NAME:DataBlock", "Name:=", "SourceNoise1", "noisedata:=", [ "FDEV=1K
"RFLO=2 ", "NCOR=2 2 " ] ],
                                [ "NAME:DataBlock", "Name:=", "SourceModulation1" "ModulationSourceData
"Butterworth", "Br=1.2288MHz", "Dly=0.5", "Iasc=1", "Qasc=1" ] ] ] ] ])
```

CopyEyeItemAsCommand

Use: Make a QuickEye copy of a VerifEye analysis, or a VerifEye copy of a QuickEye analysis.

Command: Select an analysis in the Project tree, then right-click and select **Copy As QuickEye** or **Copy As VerifEye**.

Syntax: CopyEyeItemAsCommand <itemPathList>

Return Value: Copy of the analysis.

Parameters: <itemPathList>

Type: Array of strings // Type of analysis from which to copy.

Python Syntax	CopyEyeItemAsCommand <itemPathList>

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.CopyEyeItemAsCommand( ["eye_diagram_sch|Nexxim1|Analysis|VerifEyeAnalysis"])` `oDesign.CopyEyeItemAsCommand( ["eye_diagram_sch|Nexxim1|Analysis|QuickEyeAnalysis"])`

CopyItemCommand

Use: Copy tree items, such as Altrasim Solution Setups in Nexxim, or Solve Setups and Frequency Sweeps in Ensemble.

Command: None

Syntax: CopyItemCommand <ItemPathList>

Return Value: None

Parameters: <ItemPathList>

Type: Array of strings

CreateReport

Creates a new report with a single trace and adds it to the **Results** branch in the project tree.

UI Access	Right-click on **Results** > **Create [Type] Report**
Parameters	See table below

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. "Terminal S Parameters" - Only for Driven Terminal solution-type prob-

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

			lems 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:
			`"Rectangular Plot" or "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.

	<*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.		

Python Syntax	CreateReport(<*ReportName*>, <*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*>, <*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",`
	`"OptiSetup:=", "ParametricSetup1", "SimValueContext:=",`
	`[`

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

```
                        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"]
        ],
        [
                "X Component:","Spectrum",
```


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

```
                "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

```
oModule = oDesign.GetModule("ReportSetup")
oModule.CreateReport("S Parameter Plot 4", "Modal Solution Data",
```


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

```
        "3D Rectangular Bar Plot", "Setup1 : Sweep", [],
        [
                "Freq:="        , ["All"],
                "UU:="          , ["All"],
                "ZZZ:="         , ["Nominal"]
        ],
        [
                "X Component:=", "Freq",
                "Y Component:=", "UU",
                "Z Component:=", ["dB(S(wDipole1_1_p1,wDipole1_1_p1))"]
        ])

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


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

```
E:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Filled",
                                        "Value:=", False
                                ]
                        ]
                ]
        ])

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:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Outline Width",
                                        "Value:=", "2"
                                ]
                        ]
                ]
        ])

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


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 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')`

DeleteDesignInstance

Use: Delete the design instance

Command: None

Syntax: DeleteDesignInstance <instance_name>

Return Value: None

Parameters: `<instance_name>`

Type: string

Python Syntax	DeleteDesignInstance(<instanceName>)
Python Example	`oProject.DeleteDesignInstance("Circuit1")`

DeleteOutputVariable

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

UI Access	**Circuit** > **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")`

DeletePort [Nexxim]

Use: Delete a port

Command: None

Syntax: DeletePort <Port_Name>

Return Value: None

Parameters: `<Port_Name>`

Type: string

Python Syntax	DeletePort(<portName>)
Python Example	`oDesign.DeletePort("Port1")`

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, DeleteFieldVariation, and DeleteLinked Variation.

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"]`
	`])`

DeleteSource [Nexxim]

Use: Delete the source

Command: None

Syntax: DeleteSource <srcname>

Return Value: None

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

Parameters: <srcname>

Type: string

Python Syntax	DeleteSource(<sourceName>)
Python Example	`oDesign.DeleteSource("VoltageSource1")`

EditDataset

Modifies a dataset. This can be executed by the oProject 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
      ]
    ]
]
)
```

EditImportData

Use: Edit an imported solution

Command: None

Syntax: EditImportData <oldname> <newlink> <newpath> <newname> <data> <file>

Return Value: None

Parameters: <oldname>

Type: string

<newlink>

Type: int

<newpath>

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.

<newname>

Type: string

<data>

Type: array

<file>

Type: string

EditNotes

Updates the notes for the design.

UI Access	**Circuit** > **Edit Notes**.
Parameters	Name | Type | Description *<DesignNotes>* | String | New design notes that are applied, replacing any current notes.
Return Value	None.

Python Syntax	EditNotes(*<DesignNotes>*)
Python Example	`oDesign.EditNotes("My design notes.")`

EditOutputVariable

Changes the name or expression of an existing output variable.

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.

	<*OrigVarName*>	String	Name of the original output variable
	<*NewExpression*>	String	New value to assign to the variable
	<*NewVarName*>	String	New name of the variable if any, else pass empty string
	<*SolutionName*>	String	Name of the solution as seen in the output variable UI. For example, `"Setup1 : Last Adaptive"`.
	<*ReportType*> or <*SolutionType*>	String	Name of the report type as seen in the output variable UI For Layout Editor, use the solution type.

	<*ContextArray*> or <*DomainArray*>	Array	Structured array containing context for which the output variable expression is being evaluated.
			`["Context:=", <string>]`
			For Layout Editor, use Domain array:
			`["Domain:=", <string>]`

Return Value	None		

Python Syntax	EditOutputVariable (<*OrigVarName*>, <*NewExpression*>, <*NewVarName*>, <*SolutionName*>, <*ReportType* | *SolutionType*>, <*ContextArray* | *DomainArray*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.EditOutputVariable ("test", "normalize(R1_0.V)", "testNew", "TR", "Standard", [])`

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 variables.

UI Access	**Project** > **Datasets** > **Export**.
Parameters	Name: *<datasetFileFullPath>* — Type: String — Description: The full path to the file.
Return Value	None.

Python Syntax	ExportDataset (*<datasetFileFullPath>*)
Python Example	`oProject.ExportDataset('e:/tmp/dsdata.txt')`

ExportForSpice

Exports matrix solution data to a file in a format suitable for Spice analysis. Available only for Driven Terminal solution types with ports. Output in an appropriate format will be generated for each of the non-empty file names provided.

UI Access	N/A

Parameters

Name	Type	Description
<DesignVariation>	String	Desgin variation key.
<SoluSelections>	Array	Array of selected solutions.
<SpiceType>	Integer	0 - PSpice 2 - Maxwell Spice
<BandWidth>	Integer	0 - Low (narrow) band width.
<FWSFile>	String	Full wave spice 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.

	<LumpedElementFile>	String	Lumped element file name.
	<PoleZeroSpiceFile>	String	Pole zero spice file name.
	<PoleZeroMatlabFile>	String	Pole zero Matlab file name.
	<PartialFractionFile>	String	Partial fraction file name.
	<FittingError>	Double	Optional. The accuracy to use in fitting the pole zero model.
	<MaximumOrder>	Integer	Optional. Maximum number of poles in rational function expansion.
	<UseCommonGround>	Integer	Optional. 1 to use common ground, 0 otherwise.
	<EnforcePassivity>	Integer	Optional. 1 to enforce passivity, 0 otherwise.
Return Value	None.		

Python Syntax	ExportForHSpice(*<DesignVariation>*, *<SoluSelections>*, *<SpiceType>*, *<BandWidth>*, *<FWSFile>*, *<LumpedElementFile>*, *<PoleZeroSpiceFile>*, *<PoleZeroMatlabFile>*, *<PartialFractionFile>*, [Optional *<FittingError>*], [Optional *<MaximumOrder>*], [Optional *<UseCommonGround>*], [Optional *<EnforcePassivity>*])

Python Example	`oModule.ExportForSpice("width='2in'",`
	`["Setup1:Sweep1"], 2, 0,`
	`"c:\mydir\Sweep1.fws", "", "", "", "")`

ExportNMFData

ExportNetlist [Nexxim]

Exports a netlist solution.

UI Access	None

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

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*solution*></td><td>String</td><td>The Circuit solution in the design. For Maxwell Circuit, the solution must be an empty string, "".</td></tr><tr><td><*fileName*></td><td>String</td><td>File and path to the netlist file. For Maxwell Circuit, the file name must be a SPH file.</td></tr></table>
Return Value	None.

Python Syntax	ExportNetlist(<setupName>,<fileName>)
Python Example	`oDesign.ExportNetlist("QuickEyeAnalysis", "SolutionData")`

ExportReport

Exports a report to a data file.

> **Note:**
>
> The ExportReport script command has been replaced by the script command ExportToFile. ExportReport remains in order to retain backward compatibility for existing scripts, but it is strongly recommended that you now use ExportToFile.

UI Access	None
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report to be exported</td></tr><tr><td><*FileName*></td><td>String</td><td>Full path of the exported report file name (with extension)</td></tr></table>
Return Value	None

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

Python Syntax	ExportReport(<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ExportReport(` `"Plot1", "c:\report1.dat")`

GetAllPorts

Get all ports in a design.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Ports in the design

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

GetChildNames [Design]

Returns the names of the design's child 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	Name: *<type>* — Type: String — Description: Optional. Valid options are "Module", "Editor", and "Variable". Default returns Module names.
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	Name: *<path>* — Type: String — Description: The path may include multiple generations (for example, "designObject/moduleObj/SetupObject"). See: Object Path.
Return Value	A child object if one is found. Otherwise script error.

Python Syntax	GetChildObject(*<path>*)

Python Example	Optimetrics Example:
	`oDesign = oProject.GetActiveDesign()`
	`oOptimModule = oDesign.GetChildObject("Optimetrics")`

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.GetChildObject("3D Model") oBox = oDesign.GetChildObject("3D Model/Box1") oRpt = oDesign.GetChildObject("Results/S Parameter Plot 1") oVariable= oDesign.GetChildObject("Offset")

GetChildTypes [Design]

Returns the design's child object types.

UI Access	N/A
Parameters	None.
Return Value	String: "Module", "Editor", or "Variable"

Python Syntax	GetChildTypes()
Python Example	oDesign.GetChildTypes()

GetDesignID

Returns the unique identification number of the active design.

UI Access	Not Available
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 indicating the unique identification number of the active design.

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()`

GetEditor

Use: Returns an interface to the editor requested IF the PropServer behind the PropHost is contained within that type of editor.

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

Command: None

Syntax: GetEditor(<editorName>)

Return Value: String

Python Syntax	GetEditor(<editorName>)
Python Example	`oEditor = GetEditor("SchematicEditor")`

GetLogFile

UI Access	Not Available
Parameters	Name | Type | Description <setupName> | String | Simulation setup name
Return Value	Full path to Nexxim log file, if it exists. Otherwise empty string

Python Syntax	GetLogFile()
Python Example	`fileName = oDesign.GetLogFile("Transient")`

GetModule

Returns the IDispatch for the specified module.

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

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*modulename*></td><td>String</td><td>One of the following: - Analysis Module – "AnalysisSetup" - Data Module – “DataBlock” - Optimetrics Module – "Optimetrics" - Output Variable Module – "OutputVariable" - Reporter Module – "ReportSetup" - User Defined Documents Module – "UserDefinedDocuments" - User Defined Solutions Module – "UserDefinedSolutionModule" </td></tr></table>
Return Value	Module IDispatch

Python Syntax	GetModule (<*modulename*>)
Python Example	`oModule = oDesign.GetModule("SimSetup")`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.

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 indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

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()`

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()`

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. HFSS example: `"Freq='20GHz' Theta='20deg' Phi='30deg'"`

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

	<*SolutionName*>	String	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. Possible values in Layout/Schematic Editor are: • "Standard" • "Load Pull" • "Constellation" • "Data Table" • "Eye Diagram" • "Statistical"
	<*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' Phi='30deg'", "TR", "Standard", [])`

GetOutputVariables

Returns the list of 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.

UI Access	N/A
Parameters	None.
Return Value	Array containing all output variables.

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

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A
Parameters	Name | Type | Description <*PropName*> | String | Name of the property.
Return Value	String value of the evaluated value.

Python Syntax	GetPropEvaluatedValue (<*PropName*>)
Python Example	`oVar = oDesign.GetChildObject(" Variables/var")`

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

	`oVar.GetPropEvaluatedValue()`

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Return Value: Returns interface to PropHost.

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

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. • False - do not include read only property.
Return Value	Empty array.		

Ansys 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	`oDesign.GetPropNames()`

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")`

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

	return '0.4mm' `oCreateBox.GetPropSIValue("xSize")` return 0.0004

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')`

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.

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><*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 CompInst@R;2;3</td></tr></table>
Return Value	Array of strings containing the names of the appropriate properties.

Python Syntax	GetProperties(<*PropTab*>, <*PropServer*>)
Python Example	`oEditor.GetProperties('PassedParameterTab', 'k')`

GetPropertyValue

Returns the value of a single property belonging to a specific <*PropServer*> and <*PropTab*>. This function is available with the Project, Design or Editor objects, including definition editors.

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

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

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PropTab></td><td>String</td><td>One of the following, where tab titles are shown in parentheses:
PassedParameterTab ("Parameter Values")
DefinitionParameterTab (Parameter Defaults")
LocalVariableTab ("Variables" or "Local Variables")
ProjectVariableTab ("Project variables")
ConstantsTab ("Constants")
BaseElementTab ("Symbol" or "Footprint")
ComponentTab ("General")
Component("Component")
CustomTab ("Intrinsic Variables")
Quantities ("Quantities")
Signals ("Signals")</td></tr>
<tr><td><PropServer></td><td>String</td><td>An object identifier, generally returned from another script method, such as <code>Comp: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>

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

Python Syntax	GetPropertyValue (<*PropTab*>, <*PropServer*>, <*PropName*>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for k in selectionArray:`
	`val = oEditor.GetPropertyValue("PassedParameterTab", k, "R")`
	`...`

GetResultsDirectory [Nexxim]

Gives access to the directory containing the specified Nexxim solution files.

Command: None

Syntax: GetResultsDirectory ([in] BSTR solname, [in] BSTR variation, [in] BOOL finalDir, [out, retval] BSTR* dirname)

Return Value: <dirname>

Type: String // Directory name. (The returned directory path does not contain a trailing slash.).

Parameters: <solname>

Type: string

<variation>

Type: string

<finalDir>

Type: boolean // True if the final directory is desired; False if the working/temporary directory is desired.

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

Python Syntax	dirname = GetResultsDirectory (< solname>, <variation>,<finalDir>,)
Python Example	`dirName = oDesign.GetResultsDirectory("Trans", "res='1000'", False)`

GetSourceData (Layout Editor)

Takes a source name as input and returns a VARIANT array containing a data block of the source data.

Command: None

Syntax: GetSourceData

Return Value: Type: Array of i/o block data

Parameters: `<SourceName>`

Type: string

GetVariableValue

Gets the value of a single specified variable. To get the value of project variables, execute this command using `oProject`. To get the value of local variables, use `oDesign`.

UI Access	N/A
Parameters	Name | Type | Description <*VarName*> | String | Name of the variable to access.
Return Value	String represents the value of the variable.

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

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

GetVariables

Returns a list of all defined variables. To get a list of project variables, execute this command using `oProject`. To get a list of local variables, use `oDesign`.

UI Access	N/A
Parameters	None.
Return Value	Array of strings containing the variables.

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

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.

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

	<*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')`

ImportData [Nexxim]

Imports a network solution

Command: None

Syntax: ImportData <data> <filename> <linktofile>

Return Value: None

Parameters: <data>

Type: string // data to be imported

<filename>

Type: string // name of the file

<linktofile>

Type: int // link to the file

Python Syntax	ImportData(<data>,<file>, <customization>)

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.ImportData(
   [
      "NAME:NPortData",
      "Description:="           , "",
      "ImageFile:="             , "",
      "SymbolPinConfiguration:=", 0,
      [
         "NAME:PortInfoBlk"
      ],
      [
         "NAME:PortOrderBlk"
      ],
      "filename:="      , "E:/R18Examples/Signal Integrity/models/Serial_Chan-
nel/connector_model.s4p",
      "numberofports:=" , 4,
      "sssfilename:="           , "",
      "sssmodel:="              , False,
      "PortNames:="             , ["Port1","Port2","Port3","Port4"],
      "domain:="                , "frequency",
      "datamode:="              , "Link",
      "devicename:="            , "",
```


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

```
        "SolutionName:="  , "connector_model",
        "displayformat:=" , "MagnitudePhase",
        "datatype:="              , "SMatrix",
        [
           "NAME:DesignerCustomization",
           "DCOption:="           , 0,
           "InterpOption:="       , 0,
           "ExtrapOption:="       , 1,
           "Convolution:="        , 0,
           "Passivity:="          , 0,
           "Reciprocal:="         , False,
           "ModelOption:="        , "",
           "DataType:="           , 1
        ],
        [
           "NAME:NexximCustomization",
           "DCOption:="           , 3,
           "InterpOption:="       , 1,
           "ExtrapOption:="       , 3,
           "Convolution:="                , 0,
```


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

```
        "Passivity:="          , 0,
        "Reciprocal:="         , False,
        "ModelOption:="                , "",
        "DataType:="           , 2
      ],
      [
        "NAME:HSpiceCustomization",
        "DCOption:="           , 1,
        "InterpOption:="       , 2,
        "ExtrapOption:="       , 3,
        "Convolution:="        , 0,
        "Passivity:="          , 0,
        "Reciprocal:="         , False,
        "ModelOption:="        , "",
        "DataType:="           , 3
      ],
      "NoiseModelOption:="     , "External"
  ], "", False)
```

ImportDataFilePath [Nexxim]

Returns the file path of the imported solution given an imported solution name. If the solution is not found it returns an empty string.

Command: None

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

Syntax: ImportDataFilePath <solution_name>

Return Value: File path of imported solution, empty string if solution not found in the design.

Type: string

Parameters: `<solution_name>`

Type: string

Python Syntax	ImportDataFilePath(<solution_name>)
Python Example	`solution_path_variable = oDesign.ImportDataFilePath("MySolution")`

ImportDataset

Imports a dataset from a named file. This can be executed by the oProject variables. The name of the dataset is filename+index number (e.g., dsdata1) unless the filename ends with a trailing number. When there is a trailing number at the end, we will remove the number and use first unused index. Alternatively, the name of the dataset can be explicitly defined by providing a string as an optional second argument.

UI Access	**Project** > **Datasets** > **Import**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*datasetFileFullPath*></td><td>String</td><td>The full path to the file containing the dataset values. *.tab files recommended (see note below).</td></tr><tr><td><*optionalDatasetName*></td><td>String</td><td>*Optional*. User-defined dataset name.</td></tr></table>
Return Value	None.

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

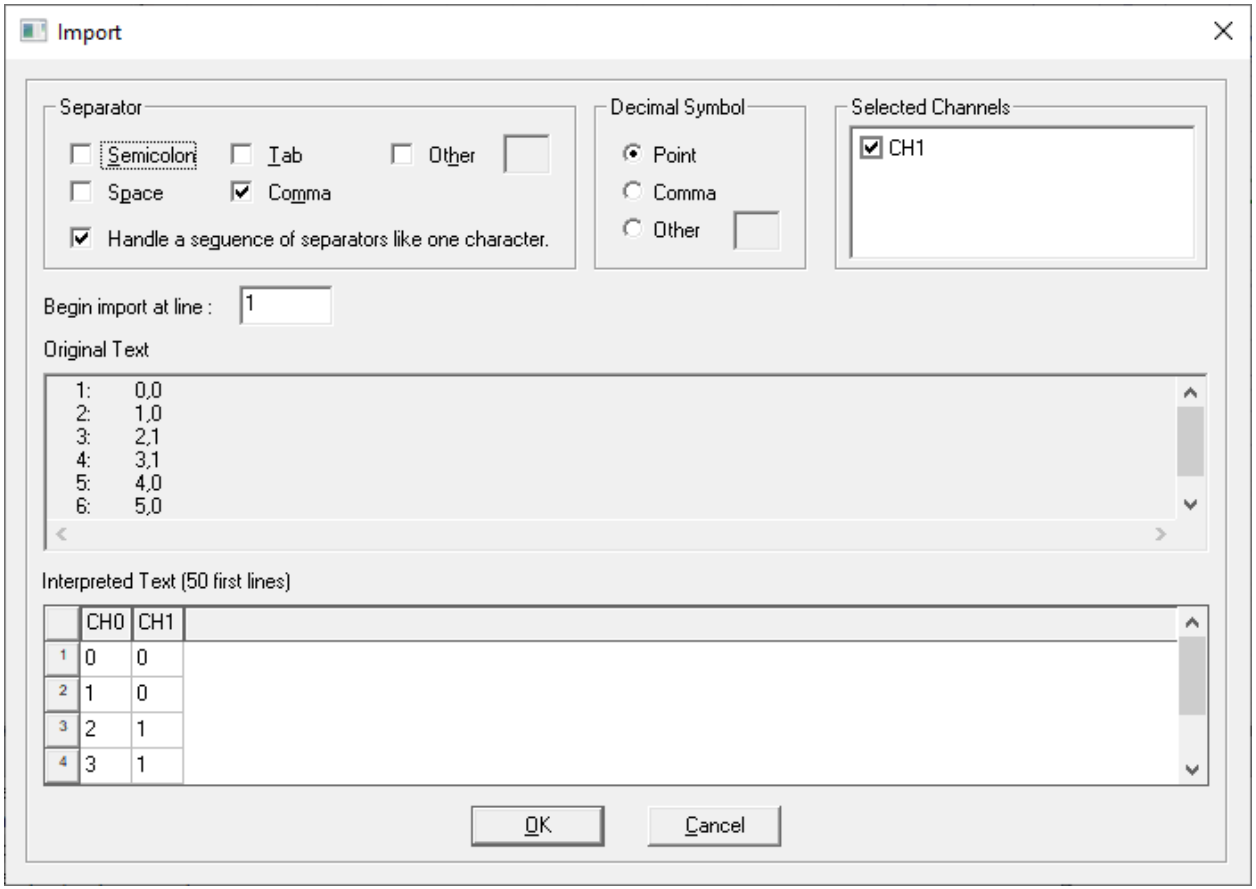
Python Syntax	ImportDataset (<*datasetFileFullPath*>,<*optionalDatasetName*>)
Python Example	`oProject.ImportDataset('e:\tmp\dsdata.tab')` `oProject.ImportDataset('e:\tmp\dsdata.tab', 'MyDatasetName')`

Note About File Types:

Tab-delimited or space-delimited files with the extension *.tab are the recommended file type. When using ImportDataset at the Design level, *.tab is the only file type supported.

At the Project level, other file types are supported (for example, *.csv). However, after calling the command, you must configure the file import format manually through the Electronics Desktop GUI by selecting **Project** > **Datasets** and clicking **Import**.

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



InsertDesign

Inserts a new design in the project. For Circuit and HFSS 3D Layout scripts, the last argument will always be empty.

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

UI Access	**Project** > **[Insert Circuit Design / Insert Circuit Netlist / Insert HFSS 3D Layout Design]**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*DesignType*></td><td>String</td><td>Design type. Valid types: "Circuit", "Nexxim Circuit", "System" , "Nexxim Netlist", "HFSS3D".</td></tr><tr><td><*DesignName*></td><td>String</td><td>Design name.</td></tr><tr><td><*TechnologyFile*></td><td>String</td><td>The path to the Circuit technology file to be used in the design. Use a pair of empty double quotes ("") for none.</td></tr><tr><td><*SubcircuitID*></td><td>String</td><td>(Optional) Must be preceded by a colon if included along with the Technology File name. No colon is necessary when the subcircuit ID is omitted.</td></tr><tr><td>""</td><td>None</td><td>Empty argument.</td></tr></table>
Return Value	None.

Python Syntax	InsertDesign (<*DesignType*>, <*DesignName*>, <*TechnologyFile*>:<*SubCircuitID*>,'')
Python Example	`oProject.InsertDesign("Nexxim Circuit","MyDesign", "C:\\Program Files\\ANSYS Inc\\v261\\AnsysEM", "")`

PasteDesign (Design Object)

Pastes a design that has already been copied to the clipboard into another design.

UI Access	**Edit** > **Paste**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*pasteOption*></td><td>Integer</td><td>One of the following: • **0** – Link to the existing design that was copied</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.

			• **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)`

PasteItemCommand

Paste tree items, such as Altrasim Solution and Solve Setups pasted to the Analysis Tree, or Frequency Sweeps pasted to the Solve Setup Tree.

Command: Right-click on the Analysis item in the Project tree and select Paste.

Syntax: PasteItemCommand <ItemPath>

Return Value: None

Parameters: `<ItemPath>`

Type: string

Python Syntax	PasteItemCommand(<ItemPath>)
Python Example	

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

	`oDesign.PasteItemCommand("Project1|Nexxim1|Analysis")`

Redo [Design]

Reapplies the last design-level command.

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

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

RemoveImportData (Layout Editor)

Remove an imported solution

Command: None

Syntax: RemoveImportData <solution_name>

Return Value: None

Parameters: `<solution_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.

RenameDesignInstance

Renames a design instance.

UI Access	Right-click a design instance in the project tree, and then click **Rename** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*OldName*></td><td>String</td><td>The current name of the design, which must be the design on which this command is invoked.</td></tr><tr><td><*NewName*></td><td>String</td><td>The new name for the design.</td></tr></table>
Return Value	None.

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

RenameImportData [Nexxim]

Rename an imported solution

Command: None

Syntax: RenameImportData <oldname> <newname>

Return Value: None

Parameters: <oldname>

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.

<newname>

Type: string

Python Syntax	RenameImportData(<oldName>,<newName>)
Python Example	`oDesign.RenameImportData("Solution", "ConnectorData")`

RenameSource [Nexxim]

Use: Rename the source

Command: None

Syntax: RenameSource <oldname>, <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Python Syntax	RenameSource(<oldName>,<newName>)
Python Example	`oDesign.RenameSource("VoltageSource1", "MySource1")`

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

RunToolkit

Runs a Python toolkit script, applying it to the active design. The toolkit script itself may have prerequisites, such as sweeps defined, or specific model characteristics, such as port definitions.

> **Important:**
>
> Full scripting for cable modeling is not supported and no arguments are allowed in the script. The **RunToolkit** command will not automatically create a cable bundle, but will simply open the **Cable Modeling** dialog box. You will have to manually input your parameters.

UI Access	[**Circuit**] > **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", "Hearing Aid Compliance", [])`

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	Name | Type | Description <*EditorName*> | String | Text of the design's notes.
Return Value	Editor object.

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

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")`

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

	`oDesign.SetPropValue("Results/S Parameter Plot 1/Display Type", "Rectangular Plot")`

SetPropertyValue

Sets the value of a single property belonging to a specific PropServer and PropTab. This function is available with the Project, Design or Editor objects, including definition editors.This is not supported for properties of the following types: ButtonProp, PointProp, V3DPointProp, and VPointProp. Only the ChangeProperty command can be used to modify these properties.

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

<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</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.

			Comp⊦nst@R;2;3
	<propName>	String	Name of the property.
	<propValue>	String	The value for the property
Return Value	None.		

Python Syntax	SetPropertyValue(*<propTab>*, *<propServer>*, *<propName>*, *<propValue>*)
Python Example	`oEditor.SetPropertyValue("PassedParameterTab", "k", "R", "2200")`

SetVariableValue

Sets the value of a variable. To set the value of a project variable, execute this command using `oProject`. To set the value of a local variable, use `oDesign`.

UI Access	N/A		
Parameters	Name	Type	Description
	<VarName>	String	Variable name.
	<VarValue>	Value	New value for the variable.
Return Value	None.		

Python Syntax	SetVariableValue (*<VarName>*, *<VarValue>*)
Python Example	`oProject.SetVariableValue('$Var1', '3mm')`

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

SimulateLink

Runs linked simulation.

UI Access	N/A
Parameters	Name | Type | Description <*setup*> | String | Setup name. <*linkedDesign*> | Object | Optional. Linked design. <*linkType*> | String | Optional. Type of linked design. <*msg*> | String | Optional. Messages to add.
Return Value	None.

Python Syntax	SimulateLink(<*setup*>, <*linkedDesign*>, <*linkType*>, <*msg*>)
Python Example	`oDesign.SimulateLink("1_4GHz")`

Undo [Design]

Cancels the last design-level command.

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

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

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

UseCircuitSParameterDefinition

Selects whether the circuit or high-frequency definition of S-parameters is used.

Command: None

Syntax: UseCircuitSparameterDefinition <boolean_operator>

Return Value: None

Parameters: `<boolean_operator>`

Type: bool

Python Syntax	UseCircuitSparameterDefinition(True)
Python Example	`oDesign.UseCircuitSparameterDefinition(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.

8 - 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

SimValueContext

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	**Circuit** > **Results** > **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.

<table>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OutputVarName></td><td>String</td><td>Name of the new output variable.</td></tr>
<tr><td><Expression></td><td>Value</td><td>Value to assign to the variable.</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution, as seen in the output variable UI.</td></tr>
<tr><td><reportType> or <solutionType></td><td>String</td><td>The name of the report type as seen in the output variable UI.

For Layout Editor, use the solution type.</td></tr>
<tr><td><ContextArray> or <DomainArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated.

<code>["Context:=", <string>]</code>

For Layout Editor, use Domain array:

<code>["Domain:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	CreateOutputVariable (<*OutputVarName*>, <*Expression*>, <*SolutionName*>, <*reportType* | *solutionType*>, <*contextArray* | *domainArray*>)
Python Example	**HFSS Example:** `oModule.CreateOutputVariable("Var" & OutputQuantity, OutputQuantity, Solution, "Far Fields", ["Context:=", InfiniteSphere, "Domain:=", "Sweep"])` **Layout Editor Example :**

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("OutputVariable") oModule.CreateOutputVariable("test","mag(S(WavePort1,WavePort1))", "Setup1 : LastAdaptive", "Modal Solution Data", ["Domain:=", "Sweep"])

DeleteOutputVariable

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

UI Access	**Circuit** > **Results** > **Output Variables**. In the **Output Variables** window, click **Delete**.
Parameters	Name | Type | Description <*OutputVarName*> | String | Name of the output variable
Return Value	None.

Python Syntax	DeleteOutputVariable (<*OutputVarName*>)
Python Example	oModule = oDesign.GetModule("OutputVariable") oModule.DeleteOutputVariable ("testNew")

DoesOutputVariableExist

Verifies whether or not a named output variable exists.

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><*outputVariableName*></td><td>String</td><td>Output variable name</td></tr></table>
Return Value	`True` if the variable exists; `False` if it does not.

Python Syntax	DoesOutputVariableExist(<*outputVariableName*>)

Python Example	`oProject = oDesktop.GetActiveProject()`
	`oDesign = oProject.GetActiveDesign()`
	`oModule = oDesign.GetModule("OutputVariable")`
	`oModule.DoesOutputVariableExist("MyTestVar")`

EditOutputVariable

Changes the name or expression of an existing output variable.

UI Access	N/A
Parameters	<table><tr><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, `"Setup1 : Last Adaptive"`.</td></tr><tr><td><*ReportType*> or <*SolutionType*></td><td>String</td><td>Name of the report type as seen in the output variable</td></tr></table>

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

			UI For Layout Editor, use the solution type.

	<*ContextArray*> or <*DomainArray*>	Array	Structured array containing context for which the output variable expression is being evaluated.
			`["Context:=", <string>]`
			For Layout Editor, use Domain array:
			`["Domain:=", <string>]`

Return Value	None		

Python Syntax	EditOutputVariable (<*OrigVarName*>, <*NewExpression*>, <*NewVarName*>, <*SolutionName*>, <*ReportType* | *SolutionType*>, <*ContextArray* | *DomainArray*>)
Python Example	`oModule = oDesign.GetModule("OutputVariable")` `oModule.EditOutputVariable ("test", "normalize(R1_0.V)", "testNew", "TR", "Standard", [])`

ExportOutputVariables

Exports output variables to a file.

UI Access	Click on **Export** in the **Output Variables** dialog.		
Parameters	Name	Type	Description
	<*FileName*>	String	Name of the file include path.
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** - output variable successfully exported. • **False** - error when export output variables.

Python Syntax	ExportOutputVariables(<*FileName*>)
Python Example	`oModule.ExportOutputVariables("C:/output_var.aoutvar")`

GetOutputVariables

Returns the list of output variables.

UI Access	N/A
Parameters	None.
Return Value	Array containing all output variables.

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

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

GetOutputVariableValue

Returns the double value of an output variable. Only expressions that return a double value are supported. The expression is evaluated only for a single point.

<table>
<tr><td>UI Access</td><td>N/A</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><OutputVarName></td><td>String</td><td>Name of the output variable.</td></tr>
<tr><td><IntrinsicVariation></td><td>String</td><td>A set of intrinsic variable value pairs to use when evaluating the output expression.
HFSS example:
<code>"Freq='20GHz' Theta='20deg' Phi='30deg'"</code></td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution as listed in the output variable UI. For example, <code>"Setup1 : Last Adaptive"</code>.</td></tr>
<tr><td><ReportType></td><td>String</td><td>The name of the report type as seen in the output variable UI.
Possible values in Layout/Schematic Editor are:
"Standard"
"Load Pull"
"Constellation"
"Data Table"
"Eye Diagram"
"Statistical"</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Structured array containing context for which the output variable expression is being evaluated. Can be empty.
<code>["Context:=", <string>]</code></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Double value of the output variable.</td></tr>
</table>

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

Python Syntax	GetOutputVariableValue(<*OutputVarName*>, <*IntrinsicVariation*>, <*SolutionName*>, <*ReportTypeName*>, <*ContextArray*>)
Python Example	`Val = oDesign.GetOutputVariableValue("test","Freq = '20Ghz' Theta='20deg' Phi='30deg'", "TR", "Standard", [])`

ImportOutputVariables

Imports output variables from a file.

UI Access	Click on **Import** in the **Output Variables** dialog.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*FileName*></td><td>String</td><td>Name of the file include path.</td></tr></table>
Return Value	Boolean: • **True** - output variable successfully imported. • **False** - error when import output variables.

Python Syntax	ImportOutputVariables(<*FileName*>)
Python Example	`oModule.ImportOutputVariables("C:/output_var.aoutvar")`

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

SimValueContext

SimValueContext holds context information for a trace, and describes how data for a trace should be extracted from the simulation. SimValueContext contains a list of 14 required initial values:

```
SimValueContext (
Domain ID, Calculation Type, Number of Cycles, Rise Time,
Step, Impulse, Context ID, Window Width,
Window Type, TDR Kaiser Parameter, Hold Time, DeviceName,
TDR Step Time, DR Maximum Time )
```

For example, the following indicates a trace in the Time Domain, Standard Calculation with the number of cycles being 2:

```
"SimValueContext:=", Array(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0)
```

Additional, context-specific values may follow the required values, as described in subsection 15 below.

1. Domain ID

No Domain	0
Time Domain	1
Spectrum Domain	2
Sweep Domain	3
Device Domain	4
SinglePt Domain	5
LoadPull Domain	6
Transient Domain	7
Budget Domain	8

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

NetworkFunction Domain	9
Oscillator Domain	55802
Noise Domain	55803
Transfer Function Domain	55807
Time Frequency Domain	55808
Transient Time Domain	55809
Periodic AC Domain	55818
UI Domain	55819
Eye Measurement Domain	55823
Initial Response Domain	55824
Peak Distortion Domain	55825

2. Calculation Type

Standard Calculation	0
Device2_DCIV	1
Device3_DCIV_Output	2
Device3_DCIV_Input	3
Device3_DCIV_Transfer	4
Device3_DCIV_Reverse	5
Device2_ACLoad	6
Device3_ACLoad_Output	7
Device3_ACLoad_Input	8

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

Device3_ACLoad_Transfer	9
Device3_ACLoad_Reverse	10
Constellation	11
EyeDiagram	12
FreeX (Statistic Report)	13

3. **Number of Cycles** — Used in Time Domain in HarmonicBalance analysis.
4. **Rise Time** — Not used by Designer/Nexxim.
5. **Step** — Not used by Designer/Nexxim.
6. **Impulse** — Not used by Designer/Nexxim.
7. **Context ID** — Not used by Designer/Nexxim.
8. **Window Width** — Not used by Designer/Nexxim.
9. **Window Type** — Not used by Designer/Nexxim.
10. **TDR Kaiser Parameter** — Not used by Designer/Nexxim.
11. **Hold Time** — Not used by Designer/Nexxim.
12. **DeviceName** — Not used by Designer/Nexxim.
13. **TDR Step Time** — Not used by Designer/Nexxim.
14. **TDR Maximum Time** — Not used by Designer/Nexxim.
15. **Context-specific values** — Used in Time Domain in HarmonicBalance analysis.

Context-specific values are entered in the format "key, true/false, keyvalue", where:

— **"key"** is the name of the key being set.

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

— **"true/false"** indicates whether the key is a string value.

— **"keyvalue"** is the value of the key.

— The order of the context keys is not significant.

— Context keys have software defaults that will be used if not provided in the script.

a. Plotting Range for Time domain in Transient and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Start Time	WS	False	0ns
Stop Time	WE	False	10ns
Minimum Time	WM	False	0ns
Maximum Time	WN	False	10ns
Is Thinning Enabled?	DE	False	0
Dy/dx Tolerance	DT	False	0.001
Number of points	DP	False	20000000

b.Transient report context for Spectral domain in Transient analysis:

Description	Key Name	Is a string?	Key Value
Start Time	TS	False	0ns

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

Stop Time	TE	False	10ns
Max Harmonics	MH	False	100
Max Frequency	MF	False	*
Window type	WT	False	0
Width Percentage	WW	False	100
Kaiser Parameter	KP	False	0
Adjust Coherent Gain	CG	False	0

* Script can specify either MH or MF. If neither is specified, Max Harmonics is set to 100. If both are specified, MF is used.

Window Type	**ID**
Rectangular	0
Bartlett	1
Blackman	2
Hamming	3
Hanning	4
Kaiser	5
Welch	6
Weber	7
Lanzcos	8

c. Eyeprobe index context for UI domain, Time domain, Eye Measuremant domain in VerifEye and QuickEye analysis:

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

Description	Key Name	Is a string?	Key Value
Eyeprobe compinst ID	PCID	False	0

d. Eyesource index context for Initial Response domain and Peak Distortion domain in VerifEye and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Eyesource compinst ID	SCID	False	0

e. UI domain context in VerifEye and QuickEye analysis:

Description	Key Name	Is a string?	Key Value
Use midpoint?	MIDPOINT	False	0 - Don't use midpoint. 1 - Use midpoint of amplitude. 2 - Use midpoint of UI.
Minimum latch overdrive	MLO	False	0

f. Distribution Context for UI Domain in VerifEye and QuickEye analysis:

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

Description	Key Name	Is a string?	Key Value
Use distribution?	USE_DIST	False	0 - No 1 - Yes
Distribution type	DIST	False	0 - Receiver Jitter 1 - Receiver Noise 2 - User Defined

Receiver Jitter Parameters

Description	Key Name	Is a string?	Key Value
DLL standard deviation	DSD	False	0
Distribution type	DIST	False	0
DLL taps	DMN	False	0
Static Offset	SOFF	False	0
Number of Gaussian data sets	NUMG	False	0
Gaussian std deviation	GS0,GS1...	False	0
Offset mean	GM1,GM1...	False	0
Number of Uniform data sets	NUMU	False	0
Uniform width	UW0,UW1...	False	0
Uniform mean	UM1,UM1...	False	0

Receiver Noise Parameters

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

Description	Key Name	Is a string?	Key Value
Number of Gaussian data sets	NUMG	False	0
Gaussian std deviation	GS0,GS1...	False	0
Number of Uniform data sets	NUMU	False	0
Uniform width	UW0,UW1...	False	0

User Defined Parameters

Description	Key Name	Is a string?	Key Value
Number of XY data pairs	NUMG	False	0
X data	X0,X1,X2...	False	0
Y data	Y0,Y1,Y2...	False	0
Cutoff probability	CP	False	0

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

9 - Reporter Editor Script Commands

Reporter commands should be executed by the oDesign object.

For example:

```
oDesign = Project.SetActiveDesign("Circuit1")

oModule = oDesign.GetModule("ReportSetup")
```

All Report and Trace properties can be edited using the **ChangeProperty** commands. This includes Title properties, General properties, and Background properties such as border color, fonts, X and Y axis scaling, and number display.

> **Note:**
>
> When you execute **Tools** > **Record Script**, operations performed in the Reporter are automatically recorded.

The list of commands is as follows:

AddAllEyeMeasurements

AddCartesianLimitLine

AddCartesianXMarker

AddCartesianYMarker

AddCartesianYMarkerToStack

AddQuickEyeAnalysis

AddDeltaMarker

AddMarker

AddNote

AddTraces

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

ClearAllMarkers

ClearAllTraceCharacteristics

CopyReportDefinitions

CopyReportData

CopyTraceDefinitions

CopyTracesData

CreateReport

CreateReportFromFile

CreateReportFromTemplate

CreateReportOfAllQuantities

DeleteAllReports

DeleteReports

DeleteTraces

ExportImageToFile

ExportToFile [Reporter]

GetAllCategories

GetAllQuantities

GetAllReportNames

GetAvailableDisplayTypes

GetAvailableReportTypes

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

GetAvailableSolutions

GetChildNames

GetChildObject

GetChildTypes

GetDisplayType

GetPropSIValue

GetPropValue [Reporter]

GetPropertyValue

GetSolutionContexts

GetSolutionDataPerVariation

GroupPlotCurvesByGroupingStrategy

ImportIntoReport

MovePlotCurvesToGroup

MovePlotCurvesToNewGroup

PasteReports

PasteTraces

RenameReport

RenameTrace

ResetPlotSettings

SavePlotSettingsAsDefault

SetPropValue

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

UpdateAllReports

UpdateReports

UpdateTraces

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.

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

	<Def>	Array	Structured array:
			`["NAME:CartesianLimitLine",`
			`  ["NAME:XValues", <integer X values>],`
			`  "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.

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>Report2D > Add Limit Line > From Selected Curve...</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 the report.</td></tr>
<tr><td><LimitLineParams></td><td>String</td><td>Structured array.
<pre>["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>]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	AddCartesianLimitLineFromCurve(<*ReportName*>, <*LimitLineParams*>)

Python Example	`oModule.AddCartesianLimitLineFromCurve("Variables Plot 2",`
	`[`
	`  "NAME:CartesianLimitLineFromCurve",`

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

```
  "TraceName:="           , "Phase",
  "CurveName:="           , "",
  "Start:="               , "0deg",
  "Stop:="                , "375deg",
  "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>)`

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

	"XValuesUnit:=", "GHz"]
Return Value	None.

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

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.
	<*MarkerName*>	String	Marker name, including any trailing number.
	<*XValue*>	Double	X coordinate.
Return Value	None		

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,`

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")`

AddCartesianYMarkerToStack

Adds a marker to a stacked report on the Y axis.

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><*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.

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	**Report2D** > **Marker** > **Add Delta Marker**.
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of report.
<*MarkerName1*>	String	Marker name, including any trailing number, for the first marker.
<*CurveName1*>	String	Full trace name for the first marker.
<*PrimarySweepValue1*>	String	Value for the primary solution.
<*MarkerName2*>	String	Marker name, including any trailing number, for the second marker.
<*CurveName2*>	String	Full trace name for the second marker.
<*PrimarySweepValue2*>	String	Value for the primary solution.

Return Value	None

Python Syntax	AddDeltaMarker(<*ReportName*>, <*MarkerName1*>, <*CurveName1*>, <*PrimarySweepValue1*>, <*MarkerName2*>, <*CurveName2*>, <*PrimarySweepValue2*>)

Python Example	`oModule.AddDeltaMarker("VerifEye Eye Plot 1", "m1",`
	`"AEYEPROBE(OutputEye) : VerifEyeAnalysis : __Amplitude=\'0.49865788706322V\' : Cartesian",`
	`"0.062", "m2",`
	`"AEYEPROBE(OutputEye) : VerifEyeAnalysis : __Amplitude=\'0.294343491063066V\' : Cartesian",`
	`"0.146")`

AddMarker

Adds a marker to a trace on a report.

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	**Report2D** > **Marker** > **Add Marker**.
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	AddMarker(<*ReportName*>, <*MarkerName*>, <*CurveName*>, <*PrimarySweepValue*>)
Python Example	`oModule.AddMarker("XY Plot 1", "m5",` `"GS1.VAL : TR4 : Cartesian", "3.61599999999997s")`

AddNote

Adds a note at a specified location to a given report.

UI Access	Right-click on the plot and select **Add Note**.
Parameters	<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><*NoteDataArray*></td><td>Array</td><td>Structured array. `["NAME:<StringDataName>", <NoteArray>]`</td></tr><tr><td><*NoteArray*></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:<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		

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."])

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

AddTraceCharacteristics

Adds a trace characteristics field to the legend on a report.

<table>
<tr><td>UI Access</td><td>Report2D > Trace Characteristics > All. This opens the Add Trace Characteristics dialog box.</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><FunctionName></td><td>String</td><td>The function name. See the Functions column of the Add Trace Characteristics dialog box.</td></tr>
<tr><td><FunctionArgs></td><td>Array</td><td>Array containing string values for any function arguments. Pass empty array if no arguments exist.

To see which argument(s) a function takes, see the bottom of the Add Trace Characteristics dialog box.

For function with one argument:

<code>[<value>]</code>

For function with multiple arguments:

<code>[<value>, <value>, ...]</code></td></tr>
<tr><td><RangeArgs></td><td>Array</td><td>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:

<code>["Specified", "19.5GHz", "24.4GHz"]</code></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	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.

UI Access	**Modify Report** > **Add Trace**.

Parameters

Name	Type	Description
<*ReportName*>	String	Name of Report
<*SolutionName*>	String	Name of the solution as listed in the **Modify Report** dialog box.

<*ContextArray*>	Array	Context for which the expression is being evaluated. This can be an empty string if there is no context.
		`["Domain:=", <DomainType>]`
		`<DomainType> ex. "Sweep" or "Time"`
		`["Context:=", <GeometryType>]`
		`<GeometryType> ex. "Infinite Sphere`*n*`", "Sphere`*n*`", "Polyline`*n*`"`

<*FamiliesArray*>	Array	Contains sweep definitions for the report.
		`["<VariableName>:= ", <ValueArray>]`
		`<ValueArray>`

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	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)"]], [])

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

ApplyReportTemplate

Applies settings to a report from a template file.

UI Access	Right-click on a report, select **Report Templates** > **Apply Settings**.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*ReportName*></td><td>String</td><td>Name of the report to apply settings to.</td></tr><tr><td><*TemplateFile*></td><td>String</td><td>Template file name with path.</td></tr><tr><td><*PropertyType*></td><td>String</td><td>Property types to apply. `("Graphical" | "Data" | "All")`</td></tr></table>
Return Value	None.

Python Syntax	ApplyReportTemplate(<*ReportName*>, <*TemplateFile*>, <*PropertyType*>)
Python Example	`oModule.ApplyReportTemplate("Variables Plot 1", "C:/MyTemplate.rpt", "Graphical")`

ChangeProperty[ReportSetup]

Change the properties for a Report.

UI Access	Double-click on a report to change its properties.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*PropertyArray*></td><td>Array</td><td>Varies, depending on the properties associated with the select object command.</td></tr></table>

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

Return Value	None.

Python Syntax	ChangeProperty(<*PropertyArray*>)

Python Example	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(
	  [
	  "NAME:AllTabs",
	  [
	  "NAME:General",
	  [
	  "NAME:PropServers",
	  "XY Plot 1:General"
	  ],
	  [
	  "NAME:ChangedProps",
	  [
	  "NAME:Use Scientific Notation",
	  "Value:=", True
	  ]
	  ]
	  ]
	])
	
	oModule = oDesign.GetModule("ReportSetup")
	  oModule.ChangeProperty(

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

```
    [
    "NAME:AllTabs",
    [
      "NAME:Legend",
      [
      "NAME:PropServers",
      "S Parameter Plot 1:Legend"
      ],
      [
      "NAME:ChangedProps",
      [
      "NAME:Legend Name",
      "Value:=" , "Ansys"
      ]
    ]
  ]
])
    oModule.ChangeProperty(
    [
      "NAME:AllTabs",
      [
        "NAME:General",
        [
          "NAME:PropServers",
          "S Parameter Plot 1:General"
        ],
    [
      "NAME:ChangedProps",
```


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

```
      [
        "NAME:Auto Scale Fonts",
        "Value:=" , False
      ]
    ]
  ]
])
```

Example For Rectangular Stacked Plot

```
oProject = oDesktop.SetActiveProject("antenna_test")
oDesign = oProject.SetActiveDesign("Design1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Scaling",
      [
        "NAME:PropServers",
        "S Parameter Plot 1:AxisY2:Stack1",
        "S Parameter Plot 1:AxisY2:Stack2"
      ],
      [
        "NAME:ChangedProps",
```


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

```
        [
          "NAME:Min",
          "Value:="               , "-60"
        ]
      ]
    ]
  ])
```

Example Change eye rect and eye plot:

```
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("test_nport_synth2")
oDesign = oProject.SetActiveDesign("Circuit1")
oModule = oDesign.GetModule("ReportSetup")
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Grid",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:Grid:Eye"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Major grid line style",
          "Value:=" , "Dot"
```


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

```
        ],
        [
          "NAME:Minor grid line style",
          "Value:=" , "Dot"
        ]
      ]
    ],
    [
      "NAME:Axis",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:AxisX:Eye"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Axis Color",
          "R:=" , 255,
          "G:=" , 128,
          "B:=" , 128
        ]
      ]
    ],
    [
      "NAME:Axis",
      [
        "NAME:PropServers",
```


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

```
          "QuickEye Voltage Plot 1:AxisY1:Rect"
        ],
        [
          "NAME:ChangedProps",
          [
            "NAME:Axis Color",
            "R:=" , 255,
            "G:=" , 128,
            "B:=" , 128
        ]
      ]
    ]
])
oModule.ChangeProperty(
  [
    "NAME:AllTabs",
    [
      "NAME:Grid",
      [
        "NAME:PropServers",
        "QuickEye Voltage Plot 1:Grid:Rect"
      ],
      [
        "NAME:ChangedProps",
        [
          "NAME:Major grid line style",
          "Value:=" , "LongDash"
        ],
```


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

	``
	          [
	            "NAME:Minor grid line style",
	            "Value:=" , "LongDash"
	          ]
	        ]
	      ]
	  ])
	``

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.

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	**Report2D** > **Trace Characteristics** > **Clear All**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of the plot</td></tr></table>
Return Value	None

Python Syntax	ClearAllTraceCharacteristics(<*PlotName*>)
Python Example	`oModule.ClearAllTraceCharacteristics("XY Plot 1")`

CloneReportsFromDatasetSolution

Clones a report for a solveed solution from a dataset solution.

UI Access	Right-click the Project tree on the report and choose **Clone from Dataset Solution** > [Dataset_SolutionName]
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportsToClone*></td><td>Array</td><td>Array of report names to clone.</td></tr><tr><td><*SoluNameToUse*></td><td>String</td><td>Name of the dataset solution.</td></tr></table>
Return Value	None.

Python Syntax	CloneReportsFromDatasetSolution(<*ReportsToClone*>, <*SoluNameToUse*>)
Python Example	`oModule.CloneReportsFromDatasetSolution(["Rectangular Plot1"], "DatasetSolution_rsm_5812")`

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

CopyPlotSettings

Copies settings of a specified plot.

UI Access	Right-click a report, select **Copy**
Parameters	Name | Type | Description <*ReportName*> | String | Name of specified report.
Return Value	None.

Python Syntax	CopyPlotSettings(<*ReportName*>)
Python Example	`oModule.CopyPlotSettings("Plot 1")`

CopyReportDefinitions

Copy the definition of a report for paste operations.

UI Access	Select a report in the Project tree, right-click and select **Copy Definition**
Parameters	Name | Type | Description <*ReportsArray*> | Array | Names of reports from which to copy the definitions
Return Value	None

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

Python Syntax	CopyReportDefinitions(<*ReportsArray*>)
Python Example	`oModule.CopyReportDefinitions(["Transmission", "Reflection"])`

CopyReportsData

Copy all data corresponding to the specified reports.

UI Access	Select a report in the Project tree, right-click and select **Copy Data**
Parameters	

Name	Type	Description
<*ReportsArray*>	Array	Names of reports from which to copy data

Return Value	None

Python Syntax	CopyReportsData (<*ReportsArray*>)
Python Example	`oModule.CopyReportsData (["Transmission","Reflection"])`

CopyTraceDefinitions

Copy trace definitions for a paste operation.

UI Access	Select a trace in the Project tree, right-click and select **Copy Definition**
Parameters	

Name	Type	Description
<*ReportName*>	String	Name of Report
<*TracesArray*>	Array	Trace definitions to copy

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

Return Value	None

Python Syntax	CopyTraceDefinitions(<*ReportName*>, <*TracesArray*>)
Python Example	`oModule.CopyTraceDefinitions ("Transmission",` `["mag(S(Port1,Port2))"])`

CopyTracesData

Copies trace data for a paste operation.

UI Access	Select a trace in the Project tree, right-click and select **Copy Data**.
Parameters	<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><*TraceArray*></td><td>Array</td><td>Trace definitions from which to copy corresponding data.</td></tr></table>
Return Value	None.

Python Syntax	CopyTracesData(<*ReportName*>, <*TracesArray*>)
Python Example	`oModule.CopyTracesData ("Transmission",` `["mag(S(Port1,Port2))"])`

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

CreateReport

Creates a new report with a single trace and adds it to the **Results** branch in the project tree.

<table>
<tr><td>UI Access</td><td>Right-click on Results > Create [Type] Report</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ReportName></td><td>String</td><td>Name of report.</td></tr>
<tr><td><ReportType></td><td>String</td><td>Type of report.
Possible values are:
"Modal S Parameters" - Only for Driven Modal solution-type problems with ports.
"Terminal S Parameters" - Only for Driven Terminal solution-type problems with ports.
"Eigenmode Parameters" - Only for Eigenmode solution-type problems.
"Fields"
"Far Fields" - Only for problems with radiation or PML boundaries.
"Near Fields" - Only for problems with radiation or PML boundaries.
"Emission Test"</td></tr>
<tr><td><DisplayType></td><td>String</td><td>Type of display.
If ReportType is "Modal S Parameters", "Terminal S Parameters", or "Eigenmode Parameters", then set to one of the following:
<code>"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Smith Chart", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot", "Rectangular Stacked Plot" or "3D Polar Plot".</code>
If <ReportType> is "Fields", then set to one of the following:</td></tr>
</table>
</td></tr>
</table>

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

			"Rectangular Plot", "Polar Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "Rectangular Stacked Plot" or "3D Rectangular Bar Plot". If <ReportType> is "Far Fields" or "Near Fields", then set to one of the following: "Rectangular Plot", "Radiation Pattern", "Rectangular Contour Plot", "Data Table", "3D Rectangular Plot", "3D Rectangular Bar Plot" "Rectangular Stacked Plot" or "3D Polar Plot" If <ReportType> is "Emission Test", then set to one of the following: "Rectangular Plot" or "Data Table"
	<SolutionName>	String	Name of the solution as listed in the **Modify Report** dialog box.
	<ContextArray>	Array	Context for which the expression is being evaluated. This can be an empty string if there is no context. ["Domain:=", <DomainType>] <DomainType> ex. "Sweep" or "Time" ["Context:=", <GeometryType>] <GeometryType> ex. "Infinite Sphere*n*", "Sphere*n*", "Polyline*n*"
	<FamiliesArray>	Array	Contains sweep definitions for the report. ["<VariableName>:= ", <ValueArray>] <ValueArray> ["All"] or ["Value1", "Value2", ..."Value*n*"]

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

			examples of <VariableName> "Freq", "Theta", "Distance"
	<ReportDataArray>	Array	This array contains the report quantity and X, Y, and (Z) axis definitions. ["X Component:=", <VariableName>, "Y Component:=", <VariableName> | <ReportQuantityArray>] <ReportQuantityArray> ex. ["dB(S(Port1, Port1))"]
Return Value	None.		

Python Syntax	CreateReport(<*ReportName*>, <*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*ContextArray*>, <*FamiliesArray*>, <*ReportDataArray*>, <*ExtendedInfo*>)

Python Example	Several examples are given below: (nominal, Optimetrics, spectral, Partial Discharge, 3D rectangular Bar Plot with Change Bar Properties, and Add a new orientation to a 3D model and 3D report)
	Nominal Example:
	`oModule.CreateReport( "XY Plot 2", "Standard",`
	`"Rectangular Plot", "TR",`
	`[`
	`"NAME:Context",`
	`"SimValueContext:=", [], 1, 0, 2, 0, false,`
	`false, -1, 1, 0, 1, 1, "", 0, 0]`
	`],`
	`[`
	`"Time:=" , ["All"],`
	`"aaa:=" , ["Nominal"]`
	`],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "X Component:=", "Time",
                "Y Component:=", ["E1.I"]
        ],
        [])
```

Optimetrics Example:

```
oModule.CreateReport ("XY Plot 3", "Standard",
        "Rectangular Plot", "TR",
        [
                "NAME:Context",
                "OptiSetup:=", "ParametricSetup1", "SimValueContext:=",
                [
                        1, 0, 2, 0, false, false, 0, 1,0, 1, 1, "", 0, 0
                ]
        ],
        [
                "Time:=", ["All"], "aaa:=", ["Nominal"]
        ],
        [
                "X Component:="  , "Time",
                "Y Component:="  , ["E1.I"]
        ],
        [])
```

Spectral Example:

```
oModule.CreateReport ("XY Plot 4", "Standard",
        "Rectangular Plot", "TR",
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NAME:Context",
                "SimValueContext:=",
                [
                        2, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "",
                        0, 0, "CG", false, "0", "KP",false, "0", "MH",
                        false, "100", "TE", false, "40ms", "TH", false,
                        "40", "TS", false, "0ns", "UF", false, "0",
                        "WT", false, "0", "WW", false, "100"
                ]
        ],
        [
                "Spectrum:="   , ["All"], "aaa:=", ["Nominal"]
        ],
        [
                "X Component:=","Spectrum",
                "Y Component:=", ["mag(E1.I)"]
        ],
        [])
```

Partial Discharge Example

```
oModule.CreateReport("Partial Discharge Plot 2",
        "Partial Discharge", "Rectangular Plot",
        "PartialDischarge1 : PartialDischarge", [],
        [
                "GasPressure:=", ["All"],
                "Freq:="       , ["All"],
                "bend_angle:=" , ["All"]
        ],
        [
                "X Component:=", "GasPressure",
                "Y Component:=", ["Power"]
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example 3D Rectangular Bar Plot with Change Bar properties

```
oModule = oDesign.GetModule("ReportSetup")
oModule.CreateReport("S Parameter Plot 4", "Modal Solution Data",
       "3D Rectangular Bar Plot", "Setup1 : Sweep", [],
       [
              "Freq:="       , ["All"],
              "UU:="         , ["All"],
              "ZZZ:="        , ["Nominal"]
       ],
       [
              "X Component:=", "Freq",
              "Y Component:=", "UU",
              "Z Component:=", ["dB(S(wDipole1_1_p1,wDipole1_1_p1))"]
       ])

oModule.ChangeProperty(
       [
              "NAME:AllTabs",
              [
                     "NAME:Bar",
                     [
                            "NAME:PropServers"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Transparency",
                                        "Value:=", "0.5"
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Filled",
                                        "Value:=", False
                                ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                [
                                        "NAME:Show Outline",
                                        "Value:=", True
                                ]
                        ]
                ]
        ])

oModule.ChangeProperty(
        [
                "NAME:AllTabs",
                [
                        "NAME:Bar",
                        [
                                "NAME:PropServers",
                                "S Parameter Plot 4:Traces-dB(S(wDipole1_1_p1,wDipole1_1_p1))"
                        ],
                        [
                                "NAME:ChangedProps",
                                [
                                        "NAME:Outline Color",
                                        "R:=", 0,
                                        "G:=", 0,
                                        "B:=", 160
                                ]
                        ]
                ]
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example Script to add a new orientation to a 3D model and 3D report

"AddViewOrientation" script command for 3D Report. The first argument must be the 3D report name.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oDesktop.OpenProject("D:/Tee.aedt")
oProject = oDesktop.SetActiveProject("Project")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.AddViewOrientation("Gain Plot 1", "3dreport", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.634171009063721,
      0.12737800180912,
      -0.762628972530365
    ],
    [
      "NAME:Up",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      -0.577874004840851,
      0.577228009700775,
      0.57694798707962
    ]
  ],
  [
    "NAME:ProjectionParams",
      [
      "NAME:Scaling",
      1.10227048397064,
      1.10227036476135,
      1.10227036476135
      ],
      [
      "NAME:Translation",
      -0.601842045783997,
      -0.352552056312561,
      -6.46677732467651
      ],
    "ProjectionType:=" , "Orthographic",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    [
      "NAME:ProjectionLimits",
      -2.53312301635742,
      2.53312301635742,
      -1,
      1,
      0.994183540344238,
      10.5401248931885
    ]
  ])
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.AddViewOrientation("3dmodel", True,
  [
  "NAME:OrientationVectorComponents",
    [
      "NAME:Direction",
      -0.627739012241364,
      -0.134194001555443,
      -0.766768991947174
    ],
    [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:Up",
      -0.694571018218994,
      0.54128497838974,
      0.473899990320206
    ]
  ],
  [
  "NAME:ProjectionParams",
    [
      "NAME:Scaling",
      0.0173665378242731,
      0.017366535961628,
      0.0173665322363377
    ],
    [
      "NAME:Translation",
      -0.0742612779140472,
      0.195189163088799,
      -5.89852476119995
    ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ProjectionType:=" , "Orthographic",
      [
        "NAME:ProjectionLimits",
        -2.27072739601135,
        2.27072739601135,
        -1,
        1,
        0.424737930297852,
        10.8078212738037
      ]
    ])
  oDesktop.CloseProject("Project")
```

CreateReportFromTemplate

Creates a report from a saved template.

UI Access	**[product]** > **Results** > **Report Templates** > **PersonalLib** > [TemplateName]
Parameters	Name | Type | Description <*TemplatePath*> | String | Path to report template
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateReportFromTemplate(<*TemplatePath*>)
Python Example	`oModule.CreateReportFromTemplate(` `"C:/MyHFSSProjects/PersonalLib/ReportTemplates/TestTemplate.rpt")`

CreateReportOfAllQuantities

Creates a report including all quantities in a category. Cannot create a report with expressions.

UI Access	NA
Parameters	

Name	Type	Description
<*ReportType*>	String	Report type name as input parameter
<*DisplayType*>	String	Display type name as input parameter
<*SolutionName*>	String	Solution name as input parameter
<*SimValueCtxt*>	String	A context name, or array of string that encoded the context(I).
<*CategoryName*>	String	Category name as input parameter
<*PointSet*>	Array	Array of strings(II)
<*CommonComponentsOfTraces*>	Array	Array of strings(III)
<*ExtTraceInfo*>	Array	Array of strings(IV)

Return Value	None.

Python Syntax	CreateReportOfAllQuantities(<*ReportNameArg*>, <*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*SimValueCtxt*>, <*CategoryName*>, <*PointSet*>, <*CommonComponentsOfTraces*>, <*ExtTraceInfo*>)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oModule.CreateReportOfAllQuantities("Smith Chart all", "Modal Solution Data", "Smith Chart", "Setup1 : LastAdaptive", [],"S Parameter",["Freq:=", ["All"], "offset:=", ["All"],"a:=", ["Nominal"],"b:=", ["Nominal"]],[],[])`

DeleteMarker

Deletes the specified marker.

UI Access	[product] > **Fields** > **Fields** > **Marker** > **Delete Marker**.
Parameters	Name | Type | Description *<MarkerName>* | String | Name of the marker.
Return Value	None.

Python Syntax	DeleteMarker(*<MarkerName>*)
Python Example	`oModule.DeleteMarker("m1")`

DeleteAllReports

Deletes all existing reports.

UI Access	Right-click the report to delete in the project tree, and then click **Delete All Reports** on the shortcut menu.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	None.

Python Syntax	DeleteAllReports()
Python Example	`oModule.DeleteAllReports()`

DeleteReports

Deletes an existing report or reports.

UI Access	Right-click the report to delete in the project tree, and then click **Delete** on the shortcut menu
Parameters	Name | Type | Description *<ReportNameArray>* | Array | Array of report names to be deleted
Return Value	None.

Python Syntax	DeleteReports(<*ReportNameArray*>)
Python Example	`oModule.DeleteReports (["Rept2DRectFreq"])`

DeleteTraceCharacteristics

Deletes a trace characteristics field from a report

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 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><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*TraceSelection*></td><td>Array</td><td>Structured array define selections. `["<ReportName>:=", <TracesArray>, <TracesArray>,... ]`</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of Report</td></tr><tr><td><*TracesArray*></td><td>Array</td><td>Contains the traces to delete within a report `[<Trace>, <Trace>, ...]`</td></tr><tr><td><*Trace*></td><td>String</td><td>A specific trace that the user wishes to delete</td></tr></table>
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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: <*DisplayType*> — Type: String — Description: Specified display type to check.
Return Value	Integer • **1** - trace characteristics is supported. • **0** - trace characteristics not supported.

Python Syntax	DoesSupportTraceCharacteristics(<*DisplayType*>)
Python Example	`oModule.DoesSupportTraceCharacteristics("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.

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. *<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	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,` `"dB() : Setup1 : Sweep1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditMarker

Edits a marker on 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 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.

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><tr><td><*ExportFileName*></td><td>String</td><td>File name to export to.</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	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	Name	Type	Description
	<*ReportName*>	String	Name of report to be exported.
	<*FileName*>	String	Full path of the exported image file name; with extension of jpg, gif, tiff, tif, bmp, or wrl.
	<*Width*>	Integer	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.
	<*Height*>	Integer	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.
	<*SaveImageParams*>> (2D)	Array	Whether to ShowLegend, ShowMarkerTable, ShowDeltaMarkerTable. Values can be True, False or Default. Default is True.
	<*SaveImageParams*> (3D)	Array	Whether to AutoFit [Default is "False"] ShowLegend, ShowMarkerTable, ShowDeltaMarkerTable. Values can be True, False or Default. Default is True.
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	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"`
	`  ])`
	`oModule.ExportImageToFile("S Parameter Plot 1", "E:/Ansoft/S Parameter Plot 1.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",
    "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"
    "ShowDeltaMarkerTable:=" , "True"
  ])
  oDesktop.CloseProject("Tee2")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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",
    "AutoFit:=" , "False",
    "Orientation:=" , "",
    "ShowOrientationGadget:=", "Default"
    "ShowLegend:=" , "True"
```


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 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:

Parameter Name	Description	Parameter Value
BackgroundType	Choose one out of four types of background	Default Plain LinearGradient RadialGradient

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
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>,`
		`"ShowGrid:=" , <string containing boolean>,`
		`"ShowRuler:=" , <string containing boolean>,`
		`"ShowRegion:=" , <string>,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"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. "Orientation:=" , <string> "ShowOrientationGadget:=" , <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	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:=" , ""])

ExportModelMeshToFile

Exports geometry model to a 3D model file (e.g. *.obj, *.wrl, etc.).

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	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>`

Type: <string>

Path to file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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.

> **Note:**
>
> The ExportReport script command has been replaced by the script command ExportToFile. ExportReport remains in order to retain backward compatibility for existing scripts, but it is strongly recommended that you now use ExportToFile.

UI Access	None		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of report to be exported
	<*FileName*>	String	Full path of the exported report file name (with extension)
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ExportReport(<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ExportReport(` `"Plot1", "c:\report1.dat")`

ExportReportDataToFile

Exports report data to a file.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of specified report. <*ExportFileName*> | String | Name of export file. File extenstion "rdat" is expected.
Return Value	None.

Python Syntax	ExportReportDataToFile(<*ReportName*>, <*ExportFileName*>)
Python Example	`oModule.ExportReportDataToFile("Plot 1", "C:/Plot1data.rdat")`

ExportTableToFile

Exports a marker table from a report to a file.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	Right-click on a plot, select **Marker** > **Export Marker Table...** or **Export Delta Marker Table...**
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>Name of export file.</td></tr><tr><td><*TableType*></td><td>String</td><td>Type of marker table to export. "Marker" or "DeltaMarker".</td></tr></table>
Return Value	None.

Python Syntax	ExportTableToFile(<*ReportName*>, <*ExportFileName*>, <*TableType*>)
Python Example	`oModule.ExportTableToFile("Plot 1", "C:/Marker.csv", "Marker")`

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	<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</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.

			.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*>)
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	Name	Type	Description
	<*ReportName*>	String	Name of the report
	<*FileName*>	String	Path and File Name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

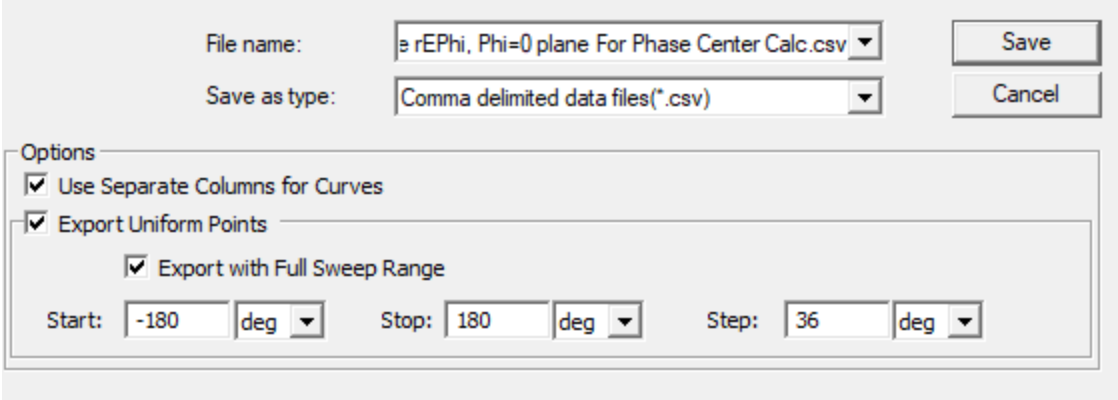
			Supported formats	
			.txt	Post processor format file
			.csv	Comma-delimited data file
			.tab	Tab-separated file
			.dat	Ansys plot data file
			.rdat	Ansys report data file
Return Value	None			

Python Syntax	ExportToFile (<*ReportName*>, <*FileName*>)
Python Example	`oModule.ExportToFile(` `"Plot1", "c:\report1.dat")`

ExportUniformPointsToFile

Exports uniform points from a data table or plot report that includes the Export Uniform Points to File option enabled to text format, comma delimited, tab delimited, or .dat type output files.

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 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 .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"

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	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

Python Syntax	GetAllCategories(<*ReportType*>, <*DisplayType*>, <*SolutionName*>, <*SimValueCtxt*>)
Python Example	`oModule.GetAllCategories("Far Fields",` `"Rectangular Plot", "Setup1 : LastAdaptive", "Infinite Sphere1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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")`

GetAllReportNames

Gets the names of existing reports in a design

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	None.
Return Value	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

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.

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 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	Name | Type | Description <*ReportType*> | String | Report type name.
Return Value	Array of text strings

Python Syntax	GetAvailableSolutions(<*ReportType*>)
Python Example	`oModule.GetAvailableSolutions("Far Fields")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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 <*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*>)

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	`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()`

GetCurvePropServerName

Gets the PropServer (the owner of the properties, or the list containing them) name of a curve.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of specified report.</td></tr><tr><td><*TraceName*></td><td>String</td><td>Name of specified trace.</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Report name</td></tr></table>
Return Value	String containing display type.

Python Syntax	GetDisplayType(<*ReportName*>)

Python Example	`oDesign = Project.SetActiveDesign("Design1")`
	`oModule = oDesign.GetModule("ReportSetup")`
	`oModule.GetDisplayType("Design Plot 1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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
Parameters	Name | Type | Description <*TraceName*> | String | Name of specified source trace. <*QtyName*> | String | Name of specified quantity value.
Return Value	Array of strings containing states.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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	Name | Type | Description <*SoluName*> | String | Name of the source solution. If empty, refer to current solution.
Return Value	Array of strings containing trace names.

Python Syntax	GetDynLinkTraces(<*SoluName*>)
Python Example	`oModule.GetDynLinkTraces("")`

GetDynLinkVariableValues

Gets the values of a variable from a dynamic linked trace.

UI Access	N/A
Parameters	Name | Type | Description <*TraceName*> | String | Name of specified dynamic linked trace, with its report name.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<*VarName*>	String	Name of specified variable.
Return Value	Array of strings containing variable values.		

Python Syntax	GetDynLinkVariableValues(<*TraceName*>, <*VarName*>)
Python Example	`oModule.GetDynLinkVariableValues("Plot 1:Trace", "Var1")`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetObjPath [Design]

Obtains the path to the design.

UI Access	N/A
Parameters	None.
Return Value	String containing the path to the design.

Python Syntax	GetObjPath()
Python Example	`oDesign.GetObjPath()`

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")` `...`

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.

Ansys 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	`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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*PlotName*></td><td>String</td><td>Name of specified plot.</td></tr></table>
Return Value	Array of strings containing trace names.

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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of a specified dumped report.</td></tr></table>
Return Value	String containing report summary.

Python Syntax	GetReportSummaryForRegressionTesting(<*ReportName*>)
Python Example	`oModule.GetReportSummaryForRegressionTesting("C:/report.rdat")`

GetSolutionContexts

Gets all available solution context names in a solution as an array of text strings.

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
<ReportType>	String	Report type name.
<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.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<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() • GetSweepUnits() • GetSweepValues() • IsPerQuantityPrimarySweep() • GetPerQuantityPrimarySweepValues() • IsDataComplex() • GetDataUnits() • GetRealDataValues() • GetImagDataValues() • ReleaseData() • GetDesignVariableNames() • GetDesignVariableUnits() • GetDesignVariableValue() • GetDesignVariationKey()		

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* recordable from the UI, but its parameters are similar to CreateReport, so you may record a CreateReport script to get the parameter values.
- For the returned ISolutionDataResultComInterface object, some of its functions have an optional boolean parameter: SIValue. SIValue defaults to True. When the pass in value is True, return data values will be in Standard International values; when False, return data values will be in the current units.

  Example: Freq Sweep with [1GHz, 2GHz,3GHz], GetSweepUnits("Freq") return "GHz"; GetSweepValues("Freq", True) return [1000000,2000000,3000000]; GetSweepValues("Freq", False) return [1,2,3].

Python Syntax	GetSolutionDataPerVariation(reportTypeArg, solutionNameArg, simValueCtxtArg, familiesArg, expressionArg)

Python Example	`oModule = oDesign.GetModule("ReportSetup")`
	
	`arr = oModule.GetSolutionDataPerVariation('Modal Solution Data', 'Setup1 : Sweep', ['Domain:=', 'Sweep'], ['Freq:=', ['All'], 'offset:=', ['All']], ['S(Port1,Port1)', 'dB(S(Port1,Port3))'])`

GetDataUnits

Returns 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.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	Name: <expressionString>; Type: String; Description: Can be returned from `GetDataExpressions()`
Return Value	Text string of units; empty if no units

Python Syntax	GetDataUnits(<expressionString>)
Python Example	`oModule.GetDataUnits(expressions)`

GetDesignVariableNames

Returns array of strings containing design variable names.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

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	NA
Return Value	Array of strings

Python Syntax	GetDesignVariableNames()
Python Example	`names = oModule.GetDesignVariableNames()`

GetDesignVariableUnits

Returns array of strings containing design variable units.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	Name | Type | Description <varName> | String | Can be returned from GetDesignVariableNames()
Return Value	Text string of units; empty if no 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	GetDesignVariableUnits(<varName>)
Python Example	`units = oModule.GetDesignVariableUnits('Variable Name')`

GetDesignVariableValue

Returns a design variable's value.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<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)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDesignVariationKey

Returns a design's Variation Key.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	String containing variation key.

Python Syntax	GetDesignVariationKey()
Python Example	`oModule.GetDesignVariationKey()`

GetImagDataValues

Returns array of imaginary data values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`</td></tr><tr><td><siValue></td><td>Integer</td><td>1 to return SI-value; 0 to return with units returned in GetSweepUnits().</td></tr></table>
Return Value	Array of doubles

Python Syntax	GetImagDataValues(<expressionString>,<siValue>)
Python Example	`imaginaryvalues = oModule.GetImagDataValues('expression',1)`

GetPerQuantityPrimarySweepValues

Returns per quantity primary sweep values.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`</td></tr><tr><td><siValue></td><td>Integer</td><td>1 to return SI-value; 0 to return by GetSweepUnits().</td></tr></table>
Return Value	Array of doubles if IsPerQuantityPrimarySweep() returned True; error if returned False

Python Syntax	GetPerQuantitySweepValues(<expressionString>, <siValue>)
Python Example	`sweepvalues = oModule.GetPerQuantitySweepValues('0.111,0.201,0.345,0.231', 1)`

GetRealDataValues

Returns array of real data values.

> **Important:**
>
> This is a member function of `ISolutionDataResultComInterface` object, which is the element of the returned array from function GetSolutionDataPerVariation.

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>

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 doubles

Python Syntax	GetRealDataValues(<expressionString>,<siValue>)
Python Example	`realvalues = oModule.GetRealDataValues('expression',1)`

GetSweepNames

Returns array of text strings containing primary sweep name(s).

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	Array of text strings

Python Syntax	GetSweepNames()
Python Example	`sweepnames = arr[0].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

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

Python Syntax	GetSweepUnits(<sweepName>)
Python Example	`sweepunits = oModule.GetSweepUnits('Sweep 1')`

GetSweepValues

Returns sweep values.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Important:

This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><sweepName></td><td>String</td><td>Primary sweep name</td></tr><tr><td><siValue></td><td>Boolean</td><td>True to return SI-value; False to return by GetSweepUnits().</td></tr></table>
Return Value	Array of doubles

Python Syntax	GetSweepValues(<sweepName>, <siValue>)
Python Example	`sweepvalues = oModule.GetSweepValues('Sweep 1', True)`

IsDataComplex

Returns whether an expression is complex.

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	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><expressionString></td><td>String</td><td>Can be returned from `GetDataExpressions()`.</td></tr></table>
Return Value	Boolean (True if expression is Complex data; False if not)

Python Syntax	IsDataComplex(<expressionString>)
Python Example	`oModule.IsDataComplex('.001,.234,.455,.434')`

IsPerQuantityPrimarySweep

Returns whether data expressions have different primary sweep values.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	Boolean (True if 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.

Python Syntax	IsPerQuantityPrimarySweep()
Python Example	`var = oModule.IsPerQuantityPrimarySweep()`

Release Data

Releases all cached data. After this function is called, all subsequent function calls to the object will fail.

> **Important:**
>
> This is a member function of ISolutionDataResultComInterface object, which is the element of the returned array from function GetSolutionDataPerVariation.

UI Access	N/A
Parameters	N/A
Return Value	N/A

Python Syntax	ReleaseData()
Python Example	`oModule.ReleaseData()`

GroupPlotCurvesByGroupingStrategy

Groups curves in a Stacked Plot automatically based on a curve grouping strategy.

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 report.</td></tr><tr><td><GroupStrategy></td><td>String</td><td>Strategy for grouping, "Single", "By Trace" or "By Units".</td></tr></table>
Return Value	None.

Python Syntax	GroupPlotCurvesByGroupingStrategy(<*ReportName*>, <*GroupStrategy*>)
Python Example	`oModule.GroupPlotCurvesByGroupingStrategy("Transient Plot 1", "By Trace")`

ImportIntoReport

Imports .tab, .csv, and .dat format files into a report.

UI Access	Right-click on report name in the Project tree and select **Import...**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	ImportIntoReport (<*ReportName*>, <*FileName*>)
Python Example	`oDesign.ImportIntoReport ("Plot1", "c:\report1.dat")`

ImportReportDataIntoReport

Imports report data from a file into a specified report.

UI Access	N/A
Parameters	Name / Type / Description <*ReportName*> / String / Name of specified report. <*FileName*> / String / Name of specified data file. File extenstion "rdat" is expected.
Return Value	None.

Python Syntax	ImportReportDataIntoReport(<*ReportName*>, <*FileName*>)
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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report.</td></tr><tr><td><*CurveArray*></td><td>Array</td><td>Array of curve names to move.</td></tr><tr><td><*StackName*></td><td>String</td><td>Name of stack to move to.</td></tr></table>
Return Value	None.

Python Syntax	MovePlotCurvesToGroup(<*ReportName*>, <*CurveArray*>, <*StackName*>)
Python Example	`oModule.MovePlotCurvesToGroup("XY Stacked Plot 1", ["R2.V : TR", "R2.I : TR"], "Stack 2")`

MovePlotCurvesToNewGroup

Move curve(s) from its stack(s) to a new stack. Here term ‘group’ is synonymous to ‘stack’ in the context of Cartesian stacked plot.

UI Access	N/A
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*ReportName*></td><td>String</td><td>Name of report.</td></tr><tr><td><*CurveArray*></td><td>Array</td><td>Array of curve names to move.</td></tr></table>
Return Value	None.

Python Syntax	MovePlotCurvesToNewGroup (<*ReportName*>, <*CurveArray*>)

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.MovePlotCurvesToNewGroup(`
	`"XY Stacked Plot 1",`
	`["R2.V : TR", "R2.I : TR"])`

OpenWindowForAllReports

Opens windows for all reports belong to current design.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	OpenWindowForAllReports()
Python Example	`oModule.OpenWindowForAllReports()`

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.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	OpenWindowForReports(<*ReportNames*>)
Python Example	`oModule.OpenWindowForReports(["Report1", "Report2"])`

PastePlotSettings

Paste plot settings to a specified report.

UI Access	Right-click a report, select **Paste**		
Parameters	Name	Type	Description
	<*ReportName*>	String	Name of specified report.
	<*PropTypeToApply*>	String	Property type to paste. "Graphical", "Data", or "All".
Return Value	None.		

Python Syntax	PastePlotSettings(<*ReportName*>, <*PropTypeToApply*>)
Python Example	`oModule.PastePlotSettings("Plot 1", "Graphical")`

PasteReports

Paste copied reports to results 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	**Edit** > **Paste**
Parameters	None.
Return Value	None.

Python Syntax	PasteReports ()
Python Example	`oModule.PasteReports()`

PasteReportsWithLegacyNames

Pastes copied reports to results in the current project with legacy name definitions.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	PasteReportsWithLegacyNames ()
Python Example	`oModule.PasteReportsWithLegacyNames()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PasteTraces

Pastes copied traces to a named plot.

UI Access	Paste
Parameters	Name | Type | Description <*ReportName*> | String | Name of plot
Return Value	None

Python Syntax	PasteTraces (<*ReportName*>)
Python Example	`oModule.PasteTraces ("XY Plot1")`

PasteTracesWithLegacyNames

Pastes copied traces to a named plot using legacy name definitions.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of 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	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.

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

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.
	<*TraceName*>	String	Name of Trace
	<*NewName*>	String	New trace name.
Return Value	None.		

Python Syntax	RenameTrace(<*ReportName*>, <*TraceName*>, <*NewName*>)
Python Example	`oModule.RenameTrace ("XY Plot1",` `"dB(S(WavePort1,WavePort1))1", "Port1dbS")`

ResetPlotSettings

Resets plot settings to defaults.

UI Access	Right-click on a plot, select **Edit** > **Reset Plot Settings**		
Parameters	Name	Type	Description
	<*PlotName*>	String	Name of specified plot.
Return Value	None.		

Python Syntax	ResetPlotSettings(<*PlotName*>)
Python Example	`oModule.ResetPlotSettings("Differential S-parameters")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SavePlotSettingsAsDefault

Saves report plot settings as default.

UI Access	**Report Templates** > **Save Settings as Default**
Parameters	Name: *<PlotName>* — Type: String — Description: Name of plot to use for plot defaults.
Return Value	None.

Python Syntax	SavePlotSettingsAsDefault ("<*PlotName*>")
Python Example	`oModule.SavePlotSettingsAsDefault("XY Plot1")`

SetLinkOutputTraces

Specifies dynamic link output traces from the current design.

UI Access	Right-click the **Results**, select **Link Output...**

Parameters	Name: *<TraceArray>* — Type: Array — Description: Array of traces to set.
	`["<ReportName>:=", <array of trace names>,`
	`"<ReportName>:=", <array of trace 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	SetLinkOutputTraces(<*TraceArray*>)

Python Example	`oModule.SetLinkOutputTraces(`
	`[`
	`  "Plot 1:=", ["Trace1"],`
	`  "Plot 2:=", ["Trace1"]`
	`])`

SetPropValue [Report Setup]

Sets the property value for report module child object.

UI Access	Select **Edit Properties** on Report objects.
Parameters	<table><tr><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*>)

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	`oRptModule.SetPropValue("S Parameter Plot 1/Display Type", "DataTable")` `oRptModule.SetPropValue("S Parameter Plot 1/db(S(port1,port2)/Primary sweep", "Freq")`

UnGroupPlotCurvesInGroup

From a Stacked Plot, ungroups curves in a stack.

UI Access	N/A
Parameters	Name | Type | Description <*ReportName*> | String | Name of report. <*GroupName*> | String | Stack group name.
Return Value	None.

Python Syntax	UnGroupPlotCurvesInGroup(<*ReportName*>, <*GroupName*>)
Python Example	`oModule.UngroupPlotCurvesInGroup("S Parameter Plot 3", "Stack 1")`

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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	UpdateAllReports()
Python Example	`oModule.UpdateAllReports()`

UpdateReports

Updates specified reports.

UI Access	N/A
Parameters	Name | Type | Description <*ReportNames*> | Array | Array of strings containing report names.
Return Value	None.

Python Syntax	UpdateReports(<*ReportNames*>)
Python Example	`oModule.UpdateReports (["XY Plot 1", "XY Plot 4"])`

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.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>In Report dialog, click Apply Traces button</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><TraceNames></td><td>Array</td><td>Array of strings containing trace names.</td></tr>
<tr><td><SolutionName></td><td>String</td><td>Name of the solution.</td></tr>
<tr><td><ContextArray></td><td>Array</td><td>Context for which the expression is being evaluated. This can be an empty string if there is no context.

["Domain:=", <DomainType>)

<DomainType> ex. "Sweep" or "Time"

["Context:=", <GeometryType>]

<GeometryType> ex. "Infinite Spheren", "Spheren", "Polylinen"</td></tr>
<tr><td><FamiliesArray></td><td>Array</td><td>Contains sweep definitions for the report.

["<VariableName>:= ", <ValueArray>]

<ValueArray>

["All"] or ["Value1", "Value2", ..."Valuen"]

examples of <VariableName> "Freq", "Theta", "Distance"</td></tr>
<tr><td><ReportDataArray></td><td>Array</td><td>This array contains the report quantity and X, Y, and (Z) axis definitions.

["X Component:=", <VariableName>,

"Y Component:=", <VariableName> | <ReportQuantityArray>]

<ReportQuantityArray> ex. ["dB(S(Port1, Port1))"]</td></tr>
<tr><td><ExtTraceInfo></td><td>Array</td><td>Optional. Array defines extended trace information.</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	UpdateTraces(<*ReportName*>, <*SolutionName*>, <*ContextArray* >, <*FamiliesArray*>, <*ReportDataArray*>)
Python Example	oModule.UpdateTraces("XY Plot 1", ["NEG1.VAL"], "TR4", ["NAME:Context", "SimValueContext:=", [2,0,2,0,False,False, -1,1,0,1,1,"",0,0,"CG",False,"0","KP",False,"0","MH",False, "100","TE",False,"100s","TH",False,"40", "TS",False,"0ns","UF",False, "0","WT",False,"0","WW",False,"100"]], ["Spectrum:=", ["All"]], ["X Component:=", "Spectrum", "Y Component:=", ["mag(NEG1.VAL)"]], [])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateTracesContextAndSweeps

Edits sweeps and context of multiple traces without affecting their component expressions.

UI Access	**Modify Report** with multiple traces selected.
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><*TraceNames*></td><td>Array</td><td>Array of strings containing trace names.</td></tr><tr><td><*SolutionName*></td><td>String</td><td>Name of the solution as listed in the **Modify Report** dialog box. For example: "`Setup1 : Last Adaptive`"</td></tr><tr><td><*ContextArray*></td><td>Array</td><td>Context for which the expression is being evaluated. This can be an empty string if there is no context. ex. "Sweep" or "Time"</td></tr><tr><td><*PointSet*></td><td>Array</td><td>Point set for the selected traces, for example, X and Y values for the plot.</td></tr></table>
Return Value	None

Python Syntax	UpdateTracesContextAndSweeps(<*ReportName*>, <*TraceNames*>, <*SolutionName*>, <*ContextArray*>, <*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",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"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.

10 - Analysis Setup Module Script Commands

Circuit analysis setup commands should be executed by the Analysis module, referred to in Circuit scripts as the "AnalysisSetup" module.

```
oModule = oDesign.GetModule("AnalysisSetup")
```

ClearLinkedData

CopySetup

DeleteDrivenSweep

DeleteSetups

EditCircuitSettings

EditSetup

ExportCircuit

GetSetupCount

GetSetups

GetSweepCount

GetSweeps

InsertFrequencySweep

InsertSetup

InsertSetup [Transient]

PasteSetup

PasteSweep

RenameDrivenSweep

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RenameSetup

RevertAllToInitial

RevertSetupToInitial

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")`

CopyDrivenSetup

Copies a driven setup.

UI Access	N/A
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	CopyDrivenSetup(<*SetupName*>)
Python Example	`oModule.CopyDrivenSetup("Setup1")`

CopyEigenSetup

Copies an eigen-analysis setup.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup.
Return Value	None.

Python Syntax	CopyEigenSetup(<*SetupName*>)
Python Example	`oModule.CopyEigenSetup("Setup1")`

CopySweep

Copies a frequency sweep from an analysis setup.

UI Access	Right-click on a sweep, select **Copy**

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 which the sweep belongs.</td></tr><tr><td><*SweepName*></td><td>String</td><td>Name of the sweep.</td></tr></table>
Return Value	None.

Python Syntax	CopySweep (<*SetupName*>, <*SweepName*>)
Python Example	`oModule.CopySweep("Setup1", "Sweep2")`

DeleteDrivenSweep

Deletes a driven frequency sweep.

UI Access	N/A
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 which the sweep belongs.</td></tr><tr><td><*SweepName*></td><td>String</td><td>Name of the sweep.</td></tr></table>
Return Value	None.

Python Syntax	DeleteDrivenSweep (<*SetupName*>, <*SweepName*>)
Python Example	`oModule.DeleteDrivenSweep("Setup1", "Sweep2")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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"])`

DoesSweepSetupExists

Determines whether a specified sweep exists.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup to which the sweep belongs. <*SweepName*> | String | Name of the sweep.
Return Value	Integer: • **1** - sweep exists. • **0** - sweep does not exist.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	DoesSweepSetupExists (<*SetupName*>, <*SweepName*>)
Python Example	`oModule.DoesSweepSetupExists("Setup1", "Sweep2")`

EditCircuitSettings

Use: Exports equivalent circuit data.

Command: Right-click the **Analysis** folder, and then choose **Edit Circuit Settings.**

Syntax: EditCircuitSettings <ExportSettings>

Return Value: None

Parameters: `<ExportSettings>`

See ExportCircuit for details.

EditDrivenSweep

Modifies a driven frequency sweep.

UI Access	N/A.
Parameters	

Name	Type	Description
<*SetupName*>	String	Name of the solution setup containing the sweep to be edited.
<*SweepName*>	String	Name of the sweep to be edited.
<*Attributes*>	Array	Structured array. `["NAME:<SweepName>",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"IsEnabled:=", <boolean>, "SetupType:=", <SetupType>, <FrequencyInformation>, "Type:=", <SweepType>, <SaveFieldsList> <DCExtrapInfo> "SMatrixOnlySolveMode:=", <"Manual">|"Auto">, "SMatrixOnlySolveAbove:=", "<real><units>"]
Return Value	None.		

Python Syntax	EditDrivenSweep (<*SetupName*>, <*SweepName*>, <*Attributes*>)
Python Example	oModule.EditDrivenSweep("Setup1", "Sweep3", ["NAME:Sweep3", "IsEnabled:=", True, "SetupType:=", "SinglePoints", "ValueList:=", ["1GHz", "2GHz", "3GHz"], "Type:=", "Discrete", "SaveFieldsList:=", [False, False, False], "ExtrapToDC:=", False])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditFrequencySweep

Modifies an existing frequency sweep. For HFSS-IE use EditSweep [HFSS-IE]

<table>
<tr><td>UI Access</td><td>Double-click a frequency sweep in the Project Manager 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 solution setup containing the sweep to be edited.</td></tr>
<tr><td><SweepName></td><td>String</td><td>Name of the sweep to be edited.</td></tr>
<tr><td><Attributes></td><td>Array</td><td>Structured array.
<pre>["NAME:<SweepName>",
  "IsEnabled:=", <boolean>,
  "SetupType:=", <SetupType>,
  <FrequencyInformation>,
  "Type:=", <SweepType>,
  <SaveFieldsList>
  <DCExtrapInfo>
  "SMatrixOnlySolveMode:=", <"Manual">|"Auto">,
  "SMatrixOnlySolveAbove:=", "<real><units>"]</pre></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None.</td></tr>
</table>

Python Syntax	EditFrequencySweep (<*SetupName*>, <*SweepName*>, <*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	oModule.EditFrequencySweep("Setup1", "Sweep3", ["NAME:Sweep3", "IsEnabled:=", True, "SetupType:=", "SinglePoints", "ValueList:=", ["1GHz", "2GHz", "3GHz"], "Type:=", "Discrete", "SaveFieldsList:=", [False, False, False], "ExtrapToDC:=", False])

EditSetup

Modifies an existing solution setup.

UI Access	Double-click a solution setup in the project tree to modify its settings.
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></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,
```


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: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",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          ["NAME:Sweep",
            "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
  ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"MaximumPasses:="       , 4,
"MinimumPasses:="       , 2,
"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"
    ]
  ]
],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "UseCacheFor:="         , ["Pass"],
  "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,
```


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,
        "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,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "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

```
oModule.EditSetup("HfssDriven",
        [
                "NAME:Setup1",
                "IsEnabled:="           , True,
                [
                        "NAME:MeshLink",
                        "ImportMesh:="          , False
                ],
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
                "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
        ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

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",
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
      "Soln:="                , "Setup1 : Sweep",
      [
        "NAME:Params",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "bend_angle:="          , "bend_angle"
      ],
      "ForceSourceToSolve:="  , True,
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "SourceProduct",
      "ApplyMeshOp:="         , True
    ],
    [
      "NAME:Excitations",
      [
        "NAME:1:1",
        "Magnitude:="           , "1",
        "Phase:="               , "0deg"
      ],
      [
        "NAME:2:1",
        "Magnitude:="           , "0",
        "Phase:="               , "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:Frequencies",
      "10GHz"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Helium"
  ])
```

ExportCircuit

Use: Export equivalent circuit data.

Command: Right-click a setup in the project tree or the **Analysis** folder and choose **Export Circuit**.

Syntax: ExportCircuit <Solution>, <Variation>, <FileName>, <ExportSettings>, <ModelName>, <Freq>

Return Value: none

Parameters: <Solution>

<SetupName>:<SolutionName>

<SetupName>

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 where circuit is being exported

<SolutionName>

Type: <String>

Name of the solution.

<Variation>

Type: <string>

The variation where circuit is being exported

<FileName>

Type: <string>

The name of the file where circuit is being exported

<ModelName>

Type: <String>

Model name or name of the sub circuit (Optional). If not specified then <FileName> is considered as model name.

<Freq>

Type: <double>

Sweep frequency in hz.

<ExportSettings>

Array("NAME:CircuitData", "MatrixName:=", _

<ReduceMatrix>, "NumberOfCells:=", <NumCell>, "UserHasChangedSettings:=", <UserChangedSettings>, "IncludeCap:=", <IncludeCap>, "IncludeCond:=", <IncludeCond>, 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:CouplingLimits", <CouplingLimitsArray>, "IncludeDCR:=", <IncludeDCR>, "IncludeDCL:=", <IncludeDCL>, "IncludeACR:=", <IncludeACR>, "IncludeACL:=", <IncludeACL>, "ADDResistance:=", <AddResistance>)

Parameters:

<ReduceMatrix>

Type: <string>

One of the reduced matrix setup or "Original"

<NumCell>

Type: <string>

Number of cells in export. Can be a variable.

<UserChangedSettings>

Type: <bool>

Whether user changed settings or use default settings.wirebond

<IncludeCap>

Type: <bool>

Flag indicates whether to export Capacitance matrix.

<IncludeCond>

Type: <bool>

Flag indicates whether to export Conductance matrix.

<IncludeDCR>

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.

```
  Flag indicates whether to export DC resistance matrix.
<IncludeDCL>
  Type: <bool>
  Flag indicates whether to export DC Inductance matrix.
<IncludeACR>
  Type: <bool>
  Flag indicates whether to export AC resistance matrix.
<IncludeACL>
  Type: <bool>
  Flag indicates whether to export AC inductance matrix.
<AddResistance>
  Type: <bool>
  Adds the DC and AC resistance.
```

Note:

You cannot export both AC and DC matrices unless **AddResistance** is selected.

```
<CouplingLimitsArray>
  Array("NAME:CouplingLimits", "CouplingLimitType:=", <CouplingLimitType>, _
  <CouplingLimitsParameters>, 0.01, "CondFraction:=", 0.01)
  <CouplingLimitType> = "None"
    Argument not needed
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<CouplingLimitType> = "ByFraction"
  <CouplingLimitsParameters>
  "CapFraction:=", <Fraction>, "IndFraction:=", <Fraction>,
  "ResFraction:=", <Fraction>,"CondFraction:=",<Fraction>,
  Parameters:
  <Fraction>
```

Type: <double>

Fraction of the self term

```
<CouplingLimitType> = "ByValue"
  <CouplingLimitsParameters>
  "CapLimit:=", <Limit>, "IndLimit:=", <Limit>, "ResLimit:=", <Limit>,
  "CondLimit:=", <Limit>,
  Parameters:
  <Limit>
  Type: <string>
  Value of the limit.
```

GetSetupCount

Gets the number of analysis setups in a design.

UI Access	N/A

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters	None.
Return Value	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 ()
Python Example	`oModule.GetSetups()`

GetSetups (Layout Editor)

Get setups.

Command: None

Syntax: GetSetups <setups>

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 array of setup names

Parameters: <setups>

Type: string

GetSetupNames

Gets the names of far field and near field radiation setups in a design.

UI Access	N/A
Parameters	Name | Type | Description <*RadiationType*> | String | Type of radiation.
Return Value	Array of strings containing setup names.

Python Syntax	GetSetupNames(<*RadiationType*>)
Python Example	`oModule.GetSetupNames("Infinite Sphere")`

GetSweepCount

Gets the number of sweeps in a specified analysis 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	Integer containing number of sweeps for the named setup.

Python Syntax	GetSweepCount (<*SetupName*>)
Python Example	`oModule.GetSweepCount("Setup1")`

GetSweeps

Gets the names of all sweeps in a given analysis setup.

UI Access	N/A
Parameters	Name | Type | Description <*SetupName*> | String | Name of specified setup.
Return Value	Array of sweep names.

Python Syntax	GetSweeps (<*SetupName*>)
Python Example	`oModule.GetSweeps("Setup1")`

GetSweeps (Layout Editor)

Get sweeps.

Command: None

Syntax: GetSweeps <sweep>

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 array of sweeps

Parameters: <sweep>

Type: string

InsertDrivenSweep

Adds a frequency sweep to a Driven solution-type setup.

UI Access	N/A.

Parameters

Name	Type	Description
<*SetupName*>	String	Name of the solution setup into which the sweep will be inserted.
<*Attributes*>	Array	Structured array. ["NAME:<string name of sweep>", "IsEnabled:=", <boolean> "RangeType:=", <"LinearStep" | "LinearCount" | "LogScale">, <LinearStepArray> "RangeStart:=", "<real>GHz", "RangeEnd:=", "<real>GHz", "RangeStep:=", "<real>GHz", <LinearCountArray> "RangeStart:=", "<real>GHz", "RangeEnd:=", "<real>GHz",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"RangeCount:=", <int>, <LogScaleArray> RangeStart:=", "<real>GHz", "RangeEnd:=", "<real>GHz", "RangeCount:=", <int>, "RangeSamples:=", <int>, <SweepTypeArray> "Type:=", <"Interpolating" |"Fast" | "Discrete">, <InterpolatingParametersArray> "SaveFields:=", <boolean>, "SaveRadFields:=", <boolean>, "InterpTolerance:=", <real>, "InterpMaxSolns:=", <int>, "InterpMinSolns:=", <int>, "InterpMinSubranges:=", <int>, "ExtrapToDC:=", <boolean>, "InterpUseS:=", <boolean>, "InterpUsePortImped:=", <boolean>, "InterpUsePropConst:=", <boolean>, "UseDerivativeConvergence:=", true, "InterpDerivTolerance:=", <real>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"UseFullBasis:=", <boolean>, "EnforcePassivity:=", <boolean>, "PassivityErrorTolerance:=", <real>) "EnforceCausality:=", <boolean>, "UseQ3DForDCSolve:=" , <boolean>, "SMatrixOnlySolveMode:=", <"Manual">|"Auto">, "SMatrixOnlySolveAbove:=", "<real><units>" <FastParametersArray> "SaveFields:=", <boolean>, "SaveRadFields:=", <boolean>, "GenerateFieldsForAllFreqs:=", <boolean>, "ExtrapToDC:=", <boolean>) <DiscreteParametersArray> "SaveFields:=", <boolean>, SaveRadFields:=", <boolean>, "ExtrapToDC:=", <boolean>]
Return Value	None.		

Python Syntax	InsertDrivenSweep(<*SetupName*>, <*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

```
oModule.InsertDrivenSweep("Setup1",
  [
    "NAME:Sweep1",
    "IsEnabled:="           , True,
    "RangeType:="           , "LinearCount",
    "RangeStart:="          , "2.5GHz",
    "RangeEnd:="            , "7.5GHz",
    "RangeCount:="          , 401,
    "Type:="                , "Interpolating",
    "SaveFields:="          , False,
    "SaveRadFields:="       , False,
    "InterpTolerance:="     , 0.5,
    "InterpMaxSolns:="      , 250,
    "InterpMinSolns:="      , 0,
    "InterpMinSubranges:="  , 1,
    "InterpUseS:="          , True,
    "InterpUsePortImped:="  , False,
    "InterpUsePropConst:="  , True,
    "UseDerivativeConvergence:=", False,
    "InterpDerivTolerance:=", 0.2,
    "UseFullBasis:="        , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "EnforcePassivity:="    , True,
    "PassivityErrorTolerance:=", 0.0001,
    "SMatrixOnlySolveMode:=", "Manual",
    "SMatrixOnlySolveAbove:=", "1MHz"
  ])
```

InsertFrequencySweep

Adds a frequency sweep to a Driven solution-type setup.

UI Access	Right-click an analysis setup, then select **Add Frequency Sweep...**.

Parameters

Name	Type	Description
<*SetupName*>	String	Name of the solution setup into which the sweep will be inserted.

<*Attributes*>	Array	Structured array.
		`["NAME:<string name of sweep>",`
		`"IsEnabled:=", <boolean>`
		`"RangeType:=", <"LinearStep" | "LinearCount" | "LogScale">,`
		`<LinearStepArray>`
		`"RangeStart:=", "<real>GHz",`
		`"RangeEnd:=", "<real>GHz",`
		`"RangeStep:=", "<real>GHz",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<LinearCountArray> "RangeStart:=", "<real>GHz", "RangeEnd:=", "<real>GHz", "RangeCount:=", <int>, <LogScaleArray> RangeStart:=", "<real>GHz", "RangeEnd:=", "<real>GHz", "RangeCount:=", <int>, "RangeSamples:=", <int>, <SweepTypeArray> "Type:=", <"Interpolating" |"Fast" | "Discrete">, <InterpolatingParametersArray> "SaveFields:=", <boolean>, "SaveRadFields:=", <boolean>, "InterpTolerance:=", <real>, "InterpMaxSolns:=", <int>, "InterpMinSolns:=", <int>, "InterpMinSubranges:=", <int>, "ExtrapToDC:=", <boolean>, "InterpUseS:=", <boolean>, "InterpUsePortImped:=", <boolean>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"InterpUsePropConst:=", <boolean>, "UseDerivativeConvergence:=", true, "InterpDerivTolerance:=", <real>, "UseFullBasis:=", <boolean>, "EnforcePassivity:=", <boolean>, "PassivityErrorTolerance:=", <real>) "EnforceCausality:=", <boolean>, "UseQ3DForDCSolve:=" , <boolean>, "SMatrixOnlySolveMode:=", <"Manual">|"Auto">, "SMatrixOnlySolveAbove:=", "<real><units>" <FastParametersArray> "SaveFields:=", <boolean>, "SaveRadFields:=", <boolean>, "GenerateFieldsForAllFreqs:=", <boolean>, "ExtrapToDC:=", <boolean>) <DiscreteParametersArray> "SaveFields:=", <boolean>, SaveRadFields:=", <boolean>, "ExtrapToDC:=", <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	InsertFrequencySweep(<*SetupName*>, <*Attributes*>)
Python Example	oModule.InsertFrequencySweep("Setup1", ["NAME:Sweep1", "IsEnabled:=" , True, "RangeType:=" , "LinearCount", "RangeStart:=" , "2.5GHz", "RangeEnd:=" , "7.5GHz", "RangeCount:=" , 401, "Type:=" , "Interpolating", "SaveFields:=" , False, "SaveRadFields:=" , False, "InterpTolerance:=" , 0.5, "InterpMaxSolns:=" , 250, "InterpMinSolns:=" , 0, "InterpMinSubranges:=" , 1, "InterpUseS:=" , True, "InterpUsePortImped:=" , False, "InterpUsePropConst:=" , True, "UseDerivativeConvergence:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "InterpDerivTolerance:=", 0.2,
    "UseFullBasis:="        , True,
    "EnforcePassivity:="    , True,
    "PassivityErrorTolerance:=", 0.0001,
    "SMatrixOnlySolveMode:=", "Manual",
    "SMatrixOnlySolveAbove:=", "1MHz"
  ])
```

InsertSetup

Adds a new solution setup.

UI Access	[product] > **Analysis Setup** > **Add Solution Setup**.
Parameters	

Name	Type	Description
<*SetupType*>	String	Type of setup to be inserted.

<*Attributes*>	Array	Structured array.
		`["NAME:<SetupName>", <NamedParameters>]`
		`<Named Parameters>`
		`"AdaptMultipleFreqs:=", <boolean>,`
		`"Frequency:=", <string>,`
		`"MaxDeltaS:=", <float>,`
		`"PortsOnly:=", <boolean>,`
		`"UseMatrixConv:=", <boolean>,`
		`### If UseMatrixCov is True, ###`
		`[`
		`"NAME:Matrix Convergence",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<Convergence Array>], MaximumPasses:=", <integer>, "MinimumPasses:=", <integer>, "MinimumConvergedPasses:=", <integer>, "PercentRefinement:=", <integer>, "IsEnabled:=", <boolean>, "BasisOrder:=", <integer>, "DoLambdaRefine:=", <boolean>, "DoMaterialLambda:=", <boolean>, "SetLambdaTarget:=", <boolean>, "Target:=", <float>, "PortAccuracy:=", <integer>, "SaveRadFieldsOnly:=", <boolean>, "SaveAnyFields:=", <boolean>, "ListsForFields:=", <array of string>, "IESolverType:=", <string>, "LambdaTargetForIESolver:=", <float> "UseDefaultLambdaTgtForIESolver:=", <boolean>
Return Value	None.		

Python Syntax	InsertSetup(<*SetupType*>, <*Attributes*>)

Python Example	Maxwell 3D AC Magnetic Setup
	`oModule = oDesign.GetModule("AnalysisSetup")`
	`oModule.InsertSetup("AC Magnetic",`
	`[`
	`"NAME:ACMagnetic_03",`
	`"Enabled:=" , True,`
	`[`
	`"NAME:MeshLink",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ImportMesh:="         , False
  ],
  "MaximumPasses:="       , 4,
  "MinimumPasses:="       , 2,
  "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"],
  "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.

```
])
```

HFSS simulation

```
oModule.InsertSetup("HfssDrivenAuto",
["NAME:Setup1",
  "Enabled:=", True,
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
            "RangeType:=", "LinearStep",
            "RangeStart:=", "1GHz",
            "RangeEnd:=", "10GHz",
            "RangeStep:=", "1GHz"
            ]
  ],
  "SaveRadFieldsOnly:=", False,
  "SaveAnyFields:=", True,
  "Type:=", "Discrete"
  ])

oModule.InsertSetup("HfssDrivenAuto",
        [
                "NAME:Setup2",
                "SolveType:="           , "Auto",
                "IsEnabled:="           , True,
        [
                "NAME:MeshLink",
                "ImportMesh:="          , False
        ],
        "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:="           , "LinearCount",
                        "RangeStart:="          , "0GHz",
                        "RangeEnd:="            , "10GHz",
                        "RangeCount:="          , 501
                ]
        ],
        "SaveRadFieldsOnly:="   , False,
        "SaveAnyFields:="       , True,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "ListsForFields:="      , ["Objectlist2"],
        "Type:="                , "Interpolating"
        ])


oModule.InsertSetup("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,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "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,
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
                                            ])
```

An HFSS with Hybrid and Arrays setup

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Project1")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "5GHz",
    "MaxDeltaE:="           , 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.

```
    "MaximumPasses:="       , 6,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "RayDensityPerWavelength:=", 4,
    "MaxNumberOfBounces:="  , 5,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "EnableSBRSelfCoupling:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    "SkipSBRSolveDuringAdaptivePasses:=", False,
    "UseSBREnhancedRadiatedPowerCalculation:=", True
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

An HFSS with Hybrid and Arrays Setup for a Component Array with FEBI Preconditioner

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.EditSetup("Fully_Coupled",
   [
     "NAME:Fully_Coupled",
     "SolveType:=" , "Single",
     "Frequency:=" , "1GHz",
     "MaxDeltaS:=" , 0.01,
     "UseMatrixConv:=" , False,
     "MaximumPasses:=" , 10,
     "MinimumPasses:=" , 1,
     "MinimumConvergedPasses:=", 1,
     "PercentRefinement:=" , 30,
     "ComponentAdaptOption:=", "Fully Coupled",
     "PerformFullDesignSolveAtLastPass:=", True,
     "IsEnabled:=" , True,
     [
       "NAME:MeshLink",
       "ImportMesh:=" , False
     ],
     "BasisOrder:=" , 1,
     "DoLambdaRefine:=" , True,
     "DoMaterialLambda:=" , True,
     "SetLambdaTarget:=" , False,
     "Target:=" , 0.3333,
     "UseMaxTetIncrease:=" , False,
     "PortAccuracy:=" , 2,
     "UseABCOnPort:=" , False,
     "SetPortMinMaxTri:=" , False,
     "DrivenSolverType:=" , "Domain Decomposition",
     "IterativeResidual:=" , 1E-06,
     "DDMSolverResidual:=" , 0.0001,
     "EnhancedLowFreqAccuracy:=", False,
     "EnhancedFEBIPreconditioner:=", True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "SaveRadFieldsOnly:=" , True,
    "SaveAnyFields:=" , True,
    "IESolverType:=" , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:=" , "Balanced",
    "InfiniteSphereSetup:=" , "3D",
    "MaxPass:=" , 10,
    "MinPass:=" , 1,
    "MinConvPass:=" , 1,
    "PerError:=" , 1,
    "PerRefine:=" , 30
  ])
```

An SBR+ Setup with Fast Frequency Looping

```
oModule.InsertSetup("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
        ])
```

SBR+ Setup with Enhanced Radiated Power Calculation

```
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup2",
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "IsSbrRangeDoppler:="   , False,
    "RayDensityPerWavelength:=", 4,
    "MaxNumberOfBounces:="  , 5,
    "EnableCWRays:="        , False,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "FastFrequencyLooping:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    [
      "NAME:Sweeps",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      [
        "NAME:Sweep",
        "RangeType:="           , "LinearStep",
        "RangeStart:="          , "1GHz",
        "RangeEnd:="            , "10GHz",
        "RangeStep:="           , "1GHz"
      ]
    ],
    "ComputeFarFields:="    , True,
    "UseSBREnhancedRadiatedPowerCalculation:=", True,
    "IsGOBlockageEnabled:=" , False,
    "GOBlockageSurfaceSelfBlock:=", False
  ])
```

Insert 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.InsertSetup("HfssRFDischarge",
  [
    "NAME:RFDischarge1",
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "Soln:="                , "Setup1 : Sweep",
      [
        "NAME:Params",
        "bend_angle:="          , "bend_angle"
      ],
      "ForceSourceToSolve:="  , False,
      "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"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Argon"
    "Save Electron Density:=", True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Is Pulsed Signal:="    , True,
    "Duty Cycle:="          , 50
  ])
```

Setup with UseMatrixConv is True and Matrix Convergence Entry Specified as All

```
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      "AllEntries:="          , True,
      "MagLimit:="            , "0.01",
      "PhaseLimit:="          , "2deg",
      "MagMinThreshold:="     , 0.01
    ],
    "MaximumPasses:="       , 12,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```

Setup with UseMatrixConv as Entries and MeshLink

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
             "NAME:Entry",
             "Port1:="          , "A[1,1]P1",
             "Port2:="          , "A[1,1]P1",
             "MagLimit:="               , "0.011",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "PhaseLimit:="            , "5deg"
        ],
        [
            "NAME:Entry",
            "Port1:="          , "A[4,4]P2",
            "Port2:="          , "A[1,1]P2",
            "MagLimit:="              , "0.041",
            "PhaseLimit:="            , "5deg"
        ]
      ]
    ],
    "MaximumPasses:="       , 1,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "",
      "Design:="              , "unit",
      "Soln:="                , "Setup1 : LastAdaptive",
      [
        "NAME:Params",
        "Airbox_dist:="         , "9.9931mm",
        "Scan_Theta:="          , "0deg",
        "Scan_phi:="            , "0deg",
        "VirtualObject_dist:="  , "2.9979mm",
        "coax_inner_rad:="      , "0.125mm",
        "coax_outer_rad:="      , "0.425mm",
        "cut_off:="             , "1.7mm",
        "feedX:="               , "-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.

```
        "feedY:="               , "2mm",
        "feed_length:="         , "1mm",
        "patchX:="              , "9.65mm",
        "rot:="                 , "45deg",
        "subH:="                , "0.8mm",
        "subX:="                , "15mm",
        "subY:="                , "15mm"
      ],
      "ForceSourceToSolve:="  , False,
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "TargetProject",
      "ApplyMeshOp:="         , True,
      "AdaptPort:="           , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , False,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Domain Decomposition",
    "IterativeResidual:="   , 0.0001,
    "DDMSolverResidual:="   , 0.0001,
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Example from design with UseMatrixConv True for AllDiagEntries

```
"UseMatrixConv:="       , True,
[
  "NAME:Matrix Convergence",
  "AllDiagEntries:="      , True,
  "DiagonalMag:="         , "0.03",
  "DiagonalPhase:="       , "4rad",
  "MagMinThreshold:="     , 0.01,
  "AllOffDiagEntries:="   , True,
  "OffDiagonalMag:="      , "0.05",
  "OffDiagonalPhase:="    , "6deg",
  "MagMinThreshold:="     , 0.01
],
```

Example from Modal design with UseMatrixConv True for Entries

```
"UseMatrixConv:="       , True,
[
  "NAME:Matrix Convergence",
  [
    "NAME:Entries",
    [
         "NAME:Entry",
         "Port1:="          , "Port1",
         "Port2:="          , "Port1",
         "MagLimit:="               , "0.02",
         "PhaseLimit:="             , "5deg"
    ],
    [
         "NAME:Entry",
         "Port1:="          , "Port3",
         "Port2:="          , "Port2",
         "MagLimit:="               , "0.03",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "PhaseLimit:="            , "5deg"
        ]
      ]
    ],
```

Example from Design with Periodic Ports and UseMaxtrixConv is True

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port1",
          "Port2:="               , "Incident_Port1",
          "MagLimit:="            , "0.11",
          "PhaseLimit:="          , "5deg"
        ],
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port2",
          "Port2:="               , "Incident_Port2",
          "MagLimit:="            , "0.22",
          "PhaseLimit:="          , "5deg"
        ]
      ]
    ],
```

Example Terminal Design with UseMatrixConv = True and Expression Cache

```
oProject = oDesktop.SetActiveProject("term-2x2_1_parasolid")
oDesign = oProject.SetActiveDesign("unit")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("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:Setup1",
"SolveType:="           , "Single",
"Frequency:="           , "10GHz",
"MaxDeltaS:="           , 0.02,
"UseMatrixConv:="       , True,
[
  "NAME:Matrix Convergence",
  [
    "NAME:Entries",
    [
         "NAME:Entry",
         "Port1:="          , "P2",
         "Port2:="          , "P1",
         "MagLimit:="               , "0.05",
         "PhaseLimit:="             , "10deg"
    ]
  ]
],
"MaximumPasses:="       , 15,
"MinimumPasses:="       , 1,
"MinimumConvergedPasses:=", 1,
"PercentRefinement:="   , 30,
"IsEnabled:="           , True,
[
  "NAME:MeshLink",
  "ImportMesh:="          , False
],
"BasisOrder:="          , 1,
"DoLambdaRefine:="      , True,
"DoMaterialLambda:="    , True,
"SetLambdaTarget:="     , False,
"Target:="              , 0.3333,
"UseMaxTetIncrease:="   , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "PortAccuracy:="        , 2,
      "UseABCOnPort:="        , False,
      "SetPortMinMaxTri:="    , False,
      "DrivenSolverType:="    , "Iterative Solver",
      "IterativeResidual:="   , 0.0001,
      "DDMSolverResidual:="   , 0.0001,
      "EnhancedLowFreqAccuracy:=", False,
      "SaveRadFieldsOnly:="   , False,
      "SaveAnyFields:="       , True,
      [
        "NAME:ExpressionCache",
        [
          "NAME:CacheItem",
          "Title:="               , "dB_AxialRatioValue_1",
          "Expression:="          , "dB(AxialRatioValue)",
          "Intrinsics:="          , "Phi=\'0deg\' Theta=\'0deg\'",
          "ReportType:="          , "Far Fields",
          [
            "NAME:ExpressionContext",
            "Context:="             , "Infinite Sphere1"
          ]
        ],
        [
          "NAME:CacheItem",
          "Title:="               , "dB_St_Diff1_Diff1__1",
          "Expression:="          , "dB(St(Diff1,Diff1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Differential Pairs"
          ]
        ],
        [
          "NAME:CacheItem",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
          "Title:="               , "dB_St_P1_P1__1",
          "Expression:="          , "dB(St(P1,P1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Terminals"
          ]
]
     ],
     "CacheSaveKind:="       , "Delta",
     "ConstantDelta:="       , "0s",
     "IESolverType:="        , "Auto",
     "LambdaTargetForIESolver:=", 0.15,
     "UseDefaultLambdaTgtForIESolver:=", True,
     "IE Solver Accuracy:="  , "Balanced",
     "InfiniteSphereSetup:=" , ""
])
```

Example HFSS Hybrid and Arrays with Beta Parallel Component Mesh Adapt

```
oModule.InsertSetup("HfssDriven",
   [
     "NAME:Setup2",
     "SolveType:="           , "Single",
     "Frequency:="           , "5GHz",
     "MaxDeltaS:="           , 0.02,
     "UseMatrixConv:="       , False,
     "MaximumPasses:="       , 6,
     "MinimumPasses:="       , 1,
     "MinimumConvergedPasses:=", 1,
     "PercentRefinement:="   , 30,
     "ComponentAdaptOption:=", "Fully Independent",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "PerformFullDesignSolveAtLastPass:=", True,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , "",
    "MaxPass:="             , 10,
    "MinPass:="             , 1,
    "MinConvPass:="         , 1,
    "PerError:="            , 1,
    "PerRefine:="           , 30
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

InsertSetup [Transient]

Adds a new solution setup to a Transient design.

Command: **HFSS** > **Analysis Setup** > **Add Solution Setup**

Syntax: InsertSetup <SetupType>, <AttributesArray>

Return Value: None

Parameters: <SetupType>

Type: <string>

"HfssTransient". Must match the HFSS solution type.

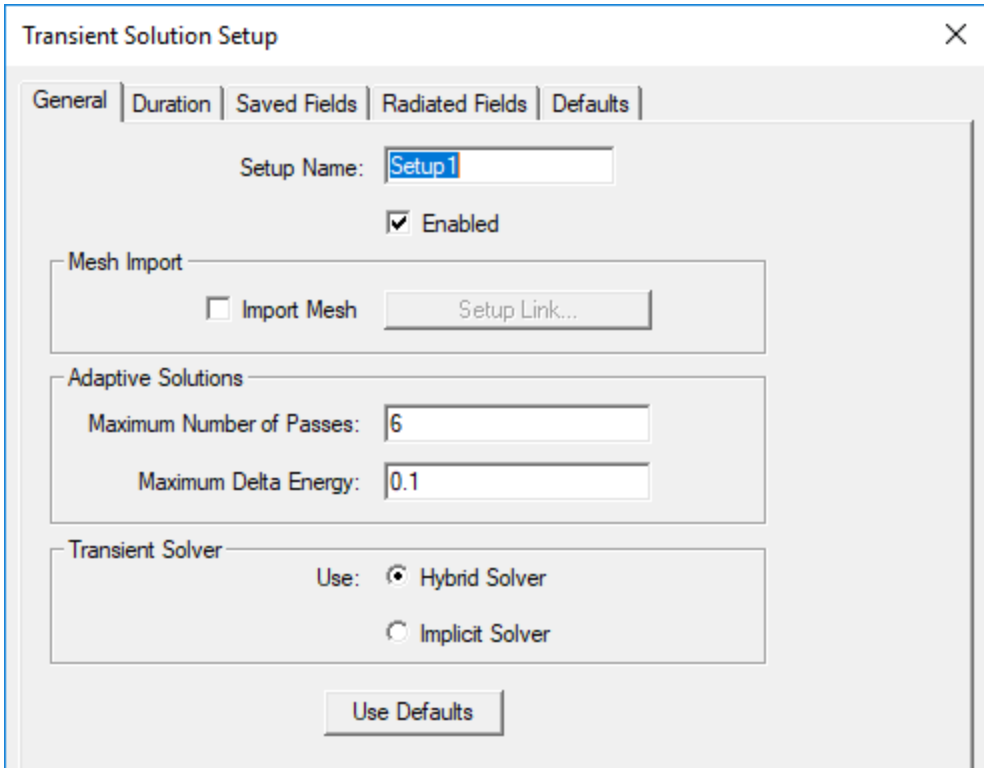
<AttributesArray>

Array("NAME:<SetupName>", <Named Parameters>)

<Named Parameters>

The parameters correspond to fields in the **Setup** dialog box. For example, here is default dialog for an HFSS Transient driven Composite Excitation setup with one excitation assigned. To see the required parameters for a specific set of parameters and their format, use the record script function, and view the resulting script in a text editor.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Python Syntax	InsertSetup("HfssTransient", [<parameters>])

Python Example	`oModule.InsertSetup("HfssTransient",`
	`  [`
	`  "NAME:Setup2",`
	`  "Frequency:=" , "5GHz",`
	`  "MaxDeltaS:=" , 0.02,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "MaximumPasses:="       , 20,
                "UseImplicitSolver:="   , True,
                "IsEnabled:="           , True,
                "BasisOrder:="          , -1,
                [
                        "NAME:Transient",
                        "TimeProfile:="         , "Broadband Pulse",
                        "HfssFrequency:="       , "5GHz",
                        "MinFreq:="             , "100MHz",
                        "MaxFreq:="             , "1GHz",
                        "NumFreqsExtracted:="   , 401,
                        "SweepMinFreq:="        , "100MHz",
                        "SweepMaxFreq:="        , "1GHz",
                        "ListsForFields:="      , ["Facelist1"],
                        "UseSaveCount:="        , 1,
                        "SaveCount:="           , 256,
                        "UseSaveStart:="        , 1,
                        "SaveStart:="           , "0s",
                        "UseSaveDelta:="        , 1,
                        "SaveDelta:="           , "10ps",
                        "SaveRadFields:="       , 0,
                        "SaveFDRadFields:="     , 0,
                        "UseAutoTermination:="  , 1,
                        "SteadyStateCriteria:=" , 0.01,
                        "UseMinimumDuration:="  , 0,
                        "TerminateOnMaximum:="  , 0
                ]
        ])
```

PasteDrivenSetup

Pastes a driven 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	N/A
Parameters	None.
Return Value	None.

Python Syntax	PasteDrivenSetup()
Python Example	`oModule.PasteDrivenSetup()`

PasteEigenSetup

Pastes an eigen-analysis setup.

UI Access	N/A
Parameters	None.
Return Value	None.

Python Syntax	PasteEigenSetup ()
Python Example	`oModule.PasteEigenSetup()`

PasteSetup

Use: Paste a solve setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: PasteSetup

Return Value: None

PasteSweep

Pastes a copied sweep.

UI Access	Right-click on a setup, select **Paste**
Parameters	Name | Type | Description <*SetupName*> | String | Name of setup where the copied sweep is pasted to.
Return Value	None.

Python Syntax	PasteSweep (<*SetupName*>)
Python Example	`oModule.PasteSweep("Setup1")`

RenameDrivenSweep

Renames an existing frequency sweep in HFSS. For HFSS-IE use RenameSweep.

UI Access	Right-click a frequency sweep in the Project Manager and then click **Rename** on the shortcut menu.
Parameters	Name | Type | Description <*SetupName*> | String | Name of setup to which the sweep belongs. <*OldSweepName*> | String | Name of the sweep to be renamed. <*NewSweepName*> | String | New name for the sweep.

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	RenameDrivenSweep (<*SetupName*>, <*OldSweepName*>, <*NewSweepName*>)
Python Example	`oModule.RenameDrivenSweep("Setup1", "Sweep1", "MySweep")`

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")`

ResetToTimeZero

Resets a simulation to time zero.

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	**CleanStop** when running Electronics Desktop in Batchmode.
Parameters	Name | Type | Description <*setupName*> | String | Name of the simulation setup to be reset.
Return Value	None.

Python Syntax	ResetToTimeZero(<*setupName*>)
Python Example	`oModule.ResetToTimeZero("Setup1")`

RevertAllToInitial

Marks the current mesh for all solution setups as invalid. This will force the next simulation to begin with the initial mesh.

UI Access	> **Analysis Setup** > **Revert to Initial Mesh**.
Parameters	None.
Return Value	None.

Python Syntax	RevertAllToInitial ()
Python Example	`oModule.RevertAllToInitial()`

RevertAllToZeroDisplacement

Reverts the displacement values of all objects to zero.

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	RevertAllToZeroDisplacement()
Python Example	`oModule.RevertAllToZeroDisplacement()`

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")`

RevertSetupToZeroDisplacement

Reverts the displacement values of objects for a specified setup to zero.

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 <*SetupName*> | String | Name of specified setup.
Return Value	None.

Python Syntax	RevertSetupToZeroDisplacement (<*SetupName*>)
Python Example	`oModule.RevertSetupToZeroDisplacement("Setup1")`

SolveSetup

Solves the specified setup.

UI Access	Right-click the setup in the project tree, and then click **Analyze** on the shortcut menu.
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup to be solved
Return Value	None

Python Syntax	SolveSetup (<*SetupName*>)
Python Example	`oModule.SolveSetup("Setup1")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

11 - Optimetrics Module Script Commands

Optimetrics script commands should be executed by the `Optimetrics` module.

```
oModule = oDesign.GetModule("Optimetrics")
oModule.CommandName <args>
```

Conventions Used in this Chapter

`<VarName>`

Type: <string>

Name of a variable.

`<VarValue>`

Type: <string>

Value with unit (i.e., <value>, but cannot be an expression).

`<StartV>`

Type: <VarValue>

The starting value of a variable.

`<StopV>`

Type: <VarValue>

The stopping value of a variable.

`<MinV>`

Type: <VarValue>

The minimum value of a variable.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<MaxV>

Type: <VarValue>

The maximum value of a variable.

<IncludeVar>

Type: <bool>

Specifies whether the variable is included in the analysis.

<StartingPoint>

```
Array("NAME:StartingPoint", "<VarName>:=",
  <VarValue>, .... "<VarName>:=", <VarValue>)
```

<SaveField>

Type: <bool>

Specifies whether HFSS will remove the non-nominal field solution.

<MaxIter>

Type: <int>

Maximum iteration allowed in an analysis.

<PriorSetup>

Type: <string>

The name of the embedded parametric setup.

<Precede>

Type: <bool>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

If true, the embedded parametric setup will be solved before the analysis begins.

If false, the embedded parametric setup will be solved during each iteration of the analysis.

```
<Constraint>
  Array("NAME:LCS",
    "lc:=", Array("<VarName>:=",
      <Coeff>, …"<VarName>:=", <Coeff>, "rel:=", <Cond>, "rhs:=", <Rhs>), ...
    "lc:=", Array("<VarName>:=", <Coeff>, ..."
      <VarName>:=", <Coeff>, "rel:=", <Cond>, "rhs:=",
      <Rhs>))
```

<Coeff>

Type: <double>

Coefficient for a variable in the linear constraint.

<Cond>

Type: <string>

Inequality condition.

<Rhs>

Type: <double>

Inequality value.

<OptiGoalSpec>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Solution:=", <Soln>, "Calculation:=", <Calc>,
   "Context:=, <Geometry>
    Array("NAME:Ranges",
     "Range:", Array("Var:=",
       <VarName>, "Type:=", <RangeType>, "Start:=",
       <StartV>, "Stop:=", <StopV>), ...
     "Range:", Array("Var:=", <VarName>, "Type:=",
       <RangeType>, "Start:=", <StartV>, "Stop:=",
       <StopV>))
```

<Soln>

Type: <string>

Name of the solution.

<Calc>

Type: <string>

An expression that is composed of a basic solution quantity and an output variable.

<ContextName>

Type: <string>

Name of context needed in the evaluation of <Calc>.

<Geometry>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: <string>

Name of geometry needed in the evaluation of <Calc>.

`<RangeType>`

Type: <string>

if "r", start and stop values specify a range for the variable.

if "s", start values specify the single value for the variable.

CopySetup

DeleteSetups [Optimetrics]

DistrbutedAnalyzeSetup

EditSetup

EditSetup [Parametric]

EditSetup [Optimization]

EditSetup [Sensitivity]

EditSetup [Statistical]

EnableSetup

ExportDXConfigFile

ExportOptimetricsProfile

ExportOptimetricsResult

ExportParametricResults

ExportParametricSetupTable

ExportRespSurfaceMinMaxTable

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportRespSurfaceRefinePoints

ExportRespSurfaceResponsePoints

ExportRespSurfaceVerificationPoints

GenerateVariationData [Parametric]

GetChildNames [Optimetrics]

GetChildObject [Optimetrics]

GetChildTypes [Optimetrics]

GetName

GetObjPath [Editor]

GetOptimetricResult

GetPropEvaluatedValue

GetPropNames [Optimetrics]

GetPropSIValue

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

ImportSetup

InsertSetup

InsertSetup [Parametric]

InsertSetup [Optimization]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

InsertSetup [Sensitivity]

InsertSetup [Statistical]

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

SolveAllSetup

SolveSetup [Optimetrics]

The topics for this section include:

General Commands Recognized by the Optimetrics Module

Parametric Script Commands

Optimization Script Commands

Sensitivity Script Commands

Statistical Script Commands

CopySetup

Copy the specified Optimetrics setup.

UI Access	NA
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup.
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CopySetup (<*SetupName*>)
Python Example	`oModule.CopySetup ("OptimizationSetup1")`

DeleteSetups [Optimetrics]

Deletes the specified Optimetrics setups.

UI Access	Right-click the setup in the project tree, and then click **Delete** on the shortcut menu
Parameters	Name | Type | Description <NameArray> | Array of Strings | An Array of Setup Names
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	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.

	<ParametricSetupName>	String	Name of the Setup
	<isBlocking>	Boolean	An optional arg that defaults to `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None		

Python Syntax	DistributedAnalyzeSetup (*<ParametricSetupName>*)
Python Example	`oModule.DistributedAnalyzeSetup("ParametricSetup1")`

EditSetup

Modifies an existing solution setup.

UI Access	Double-click a solution setup in the project tree to modify its settings.		
Parameters	Name	Type	Description
	<SetupName>	String	Name of the solve setup being edited.

	<Attributes>	Array	Structured array.
			`["NAME:<NewSetupName>", <NamedParameters>]`
			`See the InsertSetup command for details and examples.`

Return Value	None.		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditSetup (<*SetupName*>, <*Attributes*>)
Python Example	oModule.EditSetup("Setup1", ["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",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "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,
        "MinPass:=", 1,
        "MinConvPass:=", 2,
        "PerError:=", 1,
        "PerRefine:=", 30]]
)
oModule.EditSetup("HfssDrivenAuto",
["NAME:Setup1",
  "IsEnabled:=", True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
            "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",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ImportMesh:="          , False
  ],
  "MaximumPasses:="       , 4,
  "MinimumPasses:="       , 2,
  "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"
      ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ]
  ],
  "UseCacheFor:="         , ["Pass"],
  "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,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "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,
        "SetPortMinMaxTri:=", False,
        "UseDomains:=", True,
        "UseIterativeSolver:=", False,
        "IterativeResidual:=", 1E-06,
        "DDMSolverResidual:=", 0.0001,
        "EnhancedLowFreqAccuracy:=", True,
        "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,
        "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
                ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
                "RadiationSetup:="      , "",
                "PTDUTDSimulationSettings:=", "None",
                "FastFrequencyLooping:=", True,
                [
                        "NAME:Sweeps",
                        [
                                "NAME:Sweep",
                                "RangeType:="           , "LinearStep",
                                "RangeStart:="          , "1GHz",
                                "RangeEnd:="            , "10GHz",
                                "RangeStep:="           , "1GHz"
                        ]
                ],
                "ComputeFarFields:="    , True
                "UseSBREnhancedRadiatedPowerCalculation:=", True,
                "IsGOBlockageEnabled:=" , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "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",
    "Enabled:="             , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , True,
      "Project:="             , "This Project*",
      "Product:="             , "HFSS",
      "Design:="              , "This Design*",
      "Soln:="                , "Setup1 : Sweep",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:Params",
          "bend_angle:="          , "bend_angle"
        ],
        "ForceSourceToSolve:="  , True,
        "PreservePartnerSoln:=" , False,
        "PathRelativeTo:="      , "SourceProduct",
        "ApplyMeshOp:="         , True
      ],
      [
        "NAME:Excitations",
        [
          "NAME:1:1",
          "Magnitude:="           , "1",
          "Phase:="               , "0deg"
        ],
        [
          "NAME:2:1",
          "Magnitude:="           , "0",
          "Phase:="               , "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:Frequencies",
      "10GHz"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Helium"
  ])
```

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	

Name	Type	Description
<SetupName>	String	Name of the Setup
<ParametricParams>	List	List that defines the parameters of the parametric setup; examples are listed

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	below.
Return Value	None

Python Syntax	EditSetup (<SetupName>, <ParametricParams>)
Python Example	See EditSetup [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	Name	Type	Description
	<SetupName>	String	Name of the Setup
	<OptimizationParams>	List	Parameters that define the setup; examples are listed below.
Return Value	None		

Python Syntax	EditSetup (<SetupName>, <OptimizationParams>)
Python Example	oModule.EditSetup("OptimizationSetup1", ["NAME:OptimizationSetup1", "UseFastCalculationUpdateAlgo:=", False,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"FastCalcOptCtrledByUser:=", False,
"IsEnabled:=", True,
"SaveSolutions:=", False,
[
  "NAME:StartingPoint"
],
"Optimizer:=", "Quasi Newton",
[
  "NAME:AnalysisStopOptions",
  "StopForNumIteration:=" , True,
  "StopForElapsTime:=", False,
  "StopForSlowImprovement:=", False,
  "StopForGrdTolerance:=", False,
  "MaxNumIteration:=",1000,
  "MaxSolTimeInSec:=",3600,
  "RelGradientTolerance:=", 0,
  "MinNumIteration:=", 10
],
"CostFuncNormType:=", "L2",
```


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",
    "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",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    [
      "NAME:Ranges",
      "Range:=", ["Var:=", "Time","Type:=", "a"]
    ],
    "Condition:=", "==",
    [
      "NAME:GoalValue",
      "GoalValueType:=", "Independent",
      "Format:=", "Real/Imag",
      "bG:=", ["v:=", "[1;]"]
    ],
    "Weight:=", "[1;]"
  ],
  "Acceptable_Cost:="    , 0,
  "Noise:=", 0.0001,
  "UpdateDesign:=", False,
  "UpdateIteration:=", 5,
  "KeepReportAxis:=", True,
  "UpdateDesignWhenDone:=", True
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

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>
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"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
],
"Optimizer:=", "Quasi Newton",
[
"NAME:AnalysisStopOptions",
"StopForNumIteration:=", True,
"StopForElapsTime:=", False,
"StopForSlowImprovement:=", False,
"StopForGrdTolerance:="        , False,
"MaxNumIteration:=", 1001,
"MaxSolTimeInSec:=", 3600,
"RelGradientTolerance:=", 0,
"MinNumIteration:=", 10
],
"CostFuncNormType:=", "L2",
"PriorPSetup:=", "",
"PreSolvePSetup:=", True,
[
"NAME:Variables"
],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
"NAME:LCS"
],
[
"NAME:Goals",
[
"NAME:Goal",
"ReportType:=", "Standard",
"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:=", "==",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["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])

EditSetup [Statistical]

Modifies an existing 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 the setup in the project tree, and clickProperties on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr><tr><td></td><td></td><td></td></tr></table>
Return Value	None

Python Syntax	EditSetup (<SetupName>, <StatisticalParams>)
Python Example	`See EditSetup [Optimization]`

EnableSetup

Enables and disables a defined optimetrics analysis setup.

UI Access	Right-click on a setup in the project tree, select **Enable Setup** or **Disable Setup**
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SetupName*></td><td>String</td><td>Name of specified setup.</td></tr><tr><td><*Enable*></td><td>Boolean</td><td>Determines whether enable or disable a setup. **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.

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>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>
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)`

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>
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")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportRespSurfaceResponsePoints

Exports response points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Response Points** option under **View**, Click on **Export...**.
Parameters	Name | Type | Description <*DOEName*> | String | Name of the Design of Experiments (DOE) setup. <*FileName*> | String | Output file name with path.
Return Value	None.

Python Syntax	ExportRespSurfaceResponsePoints (<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceResponsePoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportRespSurfaceVerificationPoints

Exports verification points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Verification Points** option under **View**, Click on **Export...**.
Parameters	Name | Type | Description <*DOEName*> | String | Name of the Design of Experiments (DOE) setup. <*FileName*> | String | Output file name with path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ExportRespSurfaceVerificationPoints (<*DOEName*>, <*FileName*>)
Python Example	`oModule.ExportRespSurfaceVerificationPoints("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Veri_Points.csv")`

GenerateVariationData [Parametric]

Generates variation data before parametric solve for CAD integrated project.

UI Access	Right click on the parametric setup in the project tree and select **Generate Variation Data**
Parameters	Name | Type | Description <*SetupName*> | String | Name of the setup to use for the variation data generation
Return Value	None

Python Syntax	GenerateVariationData <*SetupName*>
Python Example	`oModule.GenerateVariationData("ParametricSetup1")`

GetChildNames [Optimetrics]

If used without a specific optimization setup name, gets a list of all setups for all types. If a with a specific setup name, returns names for that optimization setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	Name: typeName; Type: text string; Description: Optional, default to get all types of setup names. Or one of type name return in GetChildTypes(). Also, the type name can be used without the prefix “Opti”.
Return Value	Array of setup names.

Python Syntax	GetChildNames()

Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")`
	`arrAllSetup = oOptimModule.GetChildNames()`
	`arrParmSetup = oOptimModule.GetChildNames("'OptiParametric'")`
	`arrOptimizeSetup = oOptimModule.GetChildNames("'Optimization'")`

GetChildObject [Optimetrics]

Gets a Setup Object of the Optimetrics module

UI Access	NA
Parameters	Name: Setup Name; Type: text string; Description: A optimetrics setup name, names returned by the GetChildNames().
Return Value	A script object for the setup See discussion of Optimetrics Setup Objects in Object Script Property Function Summary.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetChildObject()
Python Example	`oParamSetup = oOptModule.GetChildObject('ParametricSetup1')` `oOptSetup = oOptModule.GetChildObject('OptimizationSetup1')`

GetChildTypes [Optimetrics]

Use: Gets child types of queried Optimetrics module.

Syntax: GetChildTypes()

Return Value: Array of text string, it can be an empty array if there is no setup is defined. There are six types of setup, they are ['OptiParametric', 'OptiOptimization', 'OptiSensitivity', 'OptiStatistical', 'OptiDesignExplorer', 'OptiDXDOE'].

Python Syntax	GetChildTypes ()
Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")` `arrSetupTypes = oOptimModule.GetChildTypes()`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1* design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetObjPath [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.

GetOptimetricResult

Returns an Optimetric calculation. The specific calculation is determined by the setup.

UI Access	N/A
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*SetupName*></td><td>String</td><td>Optimetrics setup name.</td></tr><tr><td><*vars*></td><td>Array</td><td>Array containing string variable names. Use the **Sweep Definitions** tab in the UI or the <SweepDefs> parameter in the InsertSetup script to determine appropriate inputs.</td></tr><tr><td><*values*></td><td>Array</td><td>*Optional*. Array containing string values. When multiple variables and values are provided, the order must be the same in both the <vars> and <values> arrays. The first variable is paired with the first value, the second variable is paired with the second value, and so on.</td></tr></table>
Return Value	Calculation result. If the setup contains more than one calculation, the output will be an array of values.

Python Syntax	GetOptimetricResult(<*SetupName*>, <*vars*>, <*values*>)
Python Example	`oModule.GetOptimetricResult('ParametricSetup1',['AR','Re'],['4.64','6e+04'])`

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

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	Name | Type | Description *<PropName>* | String | Name of the property.
Return Value	String value of the evaluated value.

Python Syntax	GetPropEvaluatedValue (*<PropName>*)
Python Example	`oVar = oDesign.GetChildObject(" Variables/var")` `oVar.GetPropEvaluatedValue()`

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 ()

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	`oOptModule.GetPropNames()`
	`oOptModule.GetPropNames(True)`
	`oOptModule.GetPropNames(False)`

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"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`oCreateBox.GetPropEvaluatedValue("xSize")`
	return '0.4mm'
	`oCreateBox.GetPropSIValue("xSize")`
	return 0.0004

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

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 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>)

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	`for name in oModule.GetSetupNamesByType("optimization")` `  AddInfoMessage(str(name))`

ImportSetup

Import an Optimetric setup from a file.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><SetupTypeName></td><td>String</td><td>Must be one of "OptiParametric" , "OptiOptimization", "OptiSensitivity", "OptiStatistical", or "OptiDesignExplorer".</td></tr><tr><td><SetupInfo></td><td>Array</td><td>[NAME:<SetupName>", "FilePath"] <SetupName> Type: <string>; name of the setup. <FilePath> Type : <string: file path>; must be a valid file path and name.</td></tr></table>
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.

InsertSetup

Adds a new solution setup.

<table>
<tr><td>UI Access</td><td>[product] > Analysis Setup > Add Solution Setup.</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><SetupType></td><td>String</td><td>Type of setup to be inserted.</td></tr>
<tr><td><Attributes></td><td>Array</td><td>Structured array.
<pre>["NAME:<SetupName>", <NamedParameters>]
<Named Parameters>
 "AdaptMultipleFreqs:=", <boolean>,
"Frequency:=", <string>,
"MaxDeltaS:=", <float>,
"PortsOnly:=", <boolean>,
"UseMatrixConv:=", <boolean>,
If UseMatrixCov is True,
[
"NAME:Matrix Convergence",
<Convergence Array>
],
MaximumPasses:=", <integer>,
"MinimumPasses:=", <integer>,
"MinimumConvergedPasses:=", <integer>,
"PercentRefinement:=", <integer>,
"IsEnabled:=", <boolean>,
"BasisOrder:=", <integer>,
"DoLambdaRefine:=", <boolean>,
"DoMaterialLambda:=", <boolean>,</pre></td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"SetLambdaTarget:=", <boolean>, "Target:=", <float>, "PortAccuracy:=", <integer>, "SaveRadFieldsOnly:=", <boolean>, "SaveAnyFields:=", <boolean>, "ListsForFields:=", <array of string>, "IESolverType:=", <string>, "LambdaTargetForIESolver:=", <float> "UseDefaultLambdaTgtForIESolver:=", <boolean>
Return Value	None.

Python Syntax	InsertSetup(<*SetupType*>, <*Attributes*>)
Python Example	Maxwell 3D AC Magnetic Setup oModule = oDesign.GetModule("AnalysisSetup") oModule.InsertSetup("AC Magnetic", ["NAME:ACMagnetic_03", "Enabled:=" , True, ["NAME:MeshLink", "ImportMesh:=" , False], "MaximumPasses:=" , 4, "MinimumPasses:=" , 2, "MinimumConvergedPasses:=", 1, "PercentRefinement:=" , 30, "SolveFieldOnly:=" , False, "PercentError:=" , 0.1, "SolveMatrixAtLast:=" , True, "UseNonLinearIterNum:=" , 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: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",
   "HasSweepSetup:="       , False,
   "UseHighOrderShapeFunc:=", False,
   "UseMuLink:="           , False,
   "LossAdaptiveCtrl:="    , "0.3"
])
```

HFSS simulation

```
oModule.InsertSetup("HfssDrivenAuto",
["NAME:Setup1",
  "Enabled:=", 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.InsertSetup("HfssDrivenAuto",
        [
                "NAME:Setup2",
                "SolveType:="           , "Auto",
                "IsEnabled:="           , True,
        [
                "NAME:MeshLink",
                "ImportMesh:="          , False
        ],
        "AutoSolverSetting:="   , "Balanced",
        [
                "NAME:Sweeps",
                [
                        "NAME:Sweep",
                        "RangeType:="           , "LinearCount",
                        "RangeStart:="          , "0GHz",
                        "RangeEnd:="            , "10GHz",
                        "RangeCount:="          , 501
                ]
        ],
        "SaveRadFieldsOnly:="   , False,
        "SaveAnyFields:="       , True,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "ListsForFields:="      , ["Objectlist2"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "Type:="                , "Interpolating"
        ])


oModule.InsertSetup("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,
        "SetPortMinMaxTri:=", False,
        "UseDomains:=", True,
        "UseIterativeSolver:=", False,
        "IterativeResidual:=", 1E-06,
        "DDMSolverResidual:=", 0.0001,
        "EnhancedLowFreqAccuracy:=", True,
        "SaveRadFieldsOnly:=", False,
        "SaveAnyFields:=", True,
        "IESolverType:=", "Auto",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "LambdaTargetForIESolver:=", 0.15,
        "UseDefaultLambdaTgtForIESolver:=", True,
        "SkipIERegionSolveDuringAdaptivePasses:=", True
        "RayDensityPerWavelength:=", 4,
        "MaxNumberOfBounces:="  , 5,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "SkipSBRSolveDuringAdaptivePasses:=", True,
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
                                          ])
```

An HFSS with Hybrid and Arrays setup

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("Project1")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "5GHz",
    "MaxDeltaE:="           , 0.1,
    "MaximumPasses:="       , 6,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "RayDensityPerWavelength:=", 4,
    "MaxNumberOfBounces:="  , 5,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "EnableSBRSelfCoupling:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    "SkipSBRSolveDuringAdaptivePasses:=", False,
    "UseSBREnhancedRadiatedPowerCalculation:=", True
  ])
```

An HFSS with Hybrid and Arrays Setup for a Component Array with FEBI Preconditioner

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.EditSetup("Fully_Coupled",
  [
    "NAME:Fully_Coupled",
    "SolveType:=" , "Single",
    "Frequency:=" , "1GHz",
    "MaxDeltaS:=" , 0.01,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "UseMatrixConv:=" , False,
    "MaximumPasses:=" , 10,
    "MinimumPasses:=" , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:=" , 30,
    "ComponentAdaptOption:=", "Fully Coupled",
    "PerformFullDesignSolveAtLastPass:=", True,
    "IsEnabled:=" , True,
    [
      "NAME:MeshLink",
      "ImportMesh:=" , False
    ],
    "BasisOrder:=" , 1,
    "DoLambdaRefine:=" , True,
    "DoMaterialLambda:=" , True,
    "SetLambdaTarget:=" , False,
    "Target:=" , 0.3333,
    "UseMaxTetIncrease:=" , False,
    "PortAccuracy:=" , 2,
    "UseABCOnPort:=" , False,
    "SetPortMinMaxTri:=" , False,
    "DrivenSolverType:=" , "Domain Decomposition",
    "IterativeResidual:=" , 1E-06,
    "DDMSolverResidual:=" , 0.0001,
    "EnhancedLowFreqAccuracy:=", False,
    "EnhancedFEBIPreconditioner:=", True,
    "SaveRadFieldsOnly:=" , True,
    "SaveAnyFields:=" , True,
    "IESolverType:=" , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:=" , "Balanced",
    "InfiniteSphereSetup:=" , "3D",
    "MaxPass:=" , 10,
    "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
  ])
```

An SBR+ Setup with Fast Frequency Looping

```
oModule.InsertSetup("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",
                                "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.

```
                ],
                "ComputeFarFields:="    , True
        ])
```

SBR+ Setup with Enhanced Radiated Power Calculation

```
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup2",
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "IsSbrRangeDoppler:="   , False,
    "RayDensityPerWavelength:=", 4,
    "MaxNumberOfBounces:="  , 5,
    "EnableCWRays:="        , False,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "FastFrequencyLooping:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    [
      "NAME:Sweeps",
      [
        "NAME:Sweep",
        "RangeType:="           , "LinearStep",
        "RangeStart:="          , "1GHz",
        "RangeEnd:="            , "10GHz",
        "RangeStep:="           , "1GHz"
      ]
    ],
    "ComputeFarFields:="    , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "UseSBREnhancedRadiatedPowerCalculation:=", True,
    "IsGOBlockageEnabled:=" , False,
    "GOBlockageSurfaceSelfBlock:=", False
  ])
```

Insert 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.InsertSetup("HfssRFDischarge",
  [
    "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:="  , False,
      "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",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Argon"
    "Save Electron Density:=", True,
    "Is Pulsed Signal:="    , True,
    "Duty Cycle:="          , 50
  ])
```

Setup with UseMatrixConv is True and Matrix Convergence Entry Specified as All

```
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "SolveType:="           , "Single",
      "Frequency:="           , "10GHz",
      "MaxDeltaS:="           , 0.02,
      "UseMatrixConv:="       , True,
      [
        "NAME:Matrix Convergence",
        "AllEntries:="          , True,
        "MagLimit:="            , "0.01",
        "PhaseLimit:="          , "2deg",
        "MagMinThreshold:="     , 0.01
      ],
      "MaximumPasses:="       , 12,
      "MinimumPasses:="       , 1,
      "MinimumConvergedPasses:=", 1,
      "PercentRefinement:="   , 30,
      "IsEnabled:="           , True,
      [
        "NAME:MeshLink",
        "ImportMesh:="          , False
      ],
      "BasisOrder:="          , 1,
      "DoLambdaRefine:="      , True,
      "DoMaterialLambda:="    , True,
      "SetLambdaTarget:="     , False,
      "Target:="              , 0.3333,
      "UseMaxTetIncrease:="   , False,
      "PortAccuracy:="        , 2,
      "UseABCOnPort:="        , False,
      "SetPortMinMaxTri:="    , False,
      "DrivenSolverType:="    , "Direct Solver",
      "EnhancedLowFreqAccuracy:=", 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:="        , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```

Setup with UseMatrixConv as Entries and MeshLink

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
             "NAME:Entry",
             "Port1:="          , "A[1,1]P1",
             "Port2:="          , "A[1,1]P1",
             "MagLimit:="               , "0.011",
             "PhaseLimit:="             , "5deg"
        ],
        [
             "NAME:Entry",
             "Port1:="          , "A[4,4]P2",
             "Port2:="          , "A[1,1]P2",
             "MagLimit:="               , "0.041",
             "PhaseLimit:="             , "5deg"
        ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ]
  ],
  "MaximumPasses:="       , 1,
  "MinimumPasses:="       , 1,
  "MinimumConvergedPasses:=", 1,
  "PercentRefinement:="   , 30,
  "IsEnabled:="           , True,
  [
    "NAME:MeshLink",
    "ImportMesh:="          , True,
    "Project:="             , "This Project*",
    "Product:="             , "",
    "Design:="              , "unit",
    "Soln:="                , "Setup1 : LastAdaptive",
    [
      "NAME:Params",
      "Airbox_dist:="         , "9.9931mm",
      "Scan_Theta:="          , "0deg",
      "Scan_phi:="            , "0deg",
      "VirtualObject_dist:="  , "2.9979mm",
      "coax_inner_rad:="      , "0.125mm",
      "coax_outer_rad:="      , "0.425mm",
      "cut_off:="             , "1.7mm",
      "feedX:="               , "-0.4mm",
      "feedY:="               , "2mm",
      "feed_length:="         , "1mm",
      "patchX:="              , "9.65mm",
      "rot:="                 , "45deg",
      "subH:="                , "0.8mm",
      "subX:="                , "15mm",
      "subY:="                , "15mm"
    ],
    "ForceSourceToSolve:="  , 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:=" , False,
      "PathRelativeTo:="      , "TargetProject",
      "ApplyMeshOp:="         , True,
      "AdaptPort:="           , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , False,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Domain Decomposition",
    "IterativeResidual:="   , 0.0001,
    "DDMSolverResidual:="   , 0.0001,
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```

Example from design with UseMatrixConv True for AllDiagEntries

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      "AllDiagEntries:="      , True,
      "DiagonalMag:="         , "0.03",
      "DiagonalPhase:="       , "4rad",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "MagMinThreshold:="     , 0.01,
      "AllOffDiagEntries:="   , True,
      "OffDiagonalMag:="      , "0.05",
      "OffDiagonalPhase:="    , "6deg",
      "MagMinThreshold:="     , 0.01
    ],
```

Example from Modal design with UseMatrixConv True for Entries

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
            "NAME:Entry",
            "Port1:="           , "Port1",
            "Port2:="           , "Port1",
            "MagLimit:="                , "0.02",
            "PhaseLimit:="              , "5deg"
        ],
        [
            "NAME:Entry",
            "Port1:="           , "Port3",
            "Port2:="           , "Port2",
            "MagLimit:="                , "0.03",
            "PhaseLimit:="              , "5deg"
        ]
      ]
    ],
```

Example from Design with Periodic Ports and UseMaxtrixConv is True

```
    "UseMatrixConv:="       , 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:Matrix Convergence",
      [
        "NAME:Entries",
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port1",
          "Port2:="               , "Incident_Port1",
          "MagLimit:="            , "0.11",
          "PhaseLimit:="          , "5deg"
        ],
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port2",
          "Port2:="               , "Incident_Port2",
          "MagLimit:="            , "0.22",
          "PhaseLimit:="          , "5deg"
        ]
      ]
    ],
```

Example Terminal Design with UseMatrixConv = True and Expression Cache

```
oProject = oDesktop.SetActiveProject("term-2x2_1_parasolid")
oDesign = oProject.SetActiveDesign("unit")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:Entries",
          [
               "NAME:Entry",
               "Port1:="          , "P2",
               "Port2:="          , "P1",
               "MagLimit:="               , "0.05",
               "PhaseLimit:="             , "10deg"
          ]
        ]
      ],
      "MaximumPasses:="       , 15,
      "MinimumPasses:="       , 1,
      "MinimumConvergedPasses:=", 1,
      "PercentRefinement:="   , 30,
      "IsEnabled:="           , True,
      [
        "NAME:MeshLink",
        "ImportMesh:="          , False
      ],
      "BasisOrder:="          , 1,
      "DoLambdaRefine:="      , True,
      "DoMaterialLambda:="    , True,
      "SetLambdaTarget:="     , False,
      "Target:="              , 0.3333,
      "UseMaxTetIncrease:="   , False,
      "PortAccuracy:="        , 2,
      "UseABCOnPort:="        , False,
      "SetPortMinMaxTri:="    , False,
      "DrivenSolverType:="    , "Iterative Solver",
      "IterativeResidual:="   , 0.0001,
      "DDMSolverResidual:="   , 0.0001,
      "EnhancedLowFreqAccuracy:=", False,
```


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,
      [
        "NAME:ExpressionCache",
        [
          "NAME:CacheItem",
          "Title:="               , "dB_AxialRatioValue_1",
          "Expression:="          , "dB(AxialRatioValue)",
          "Intrinsics:="          , "Phi=\'0deg\' Theta=\'0deg\'",
          "ReportType:="          , "Far Fields",
          [
            "NAME:ExpressionContext",
            "Context:="             , "Infinite Sphere1"
          ]
        ],
        [
          "NAME:CacheItem",
          "Title:="               , "dB_St_Diff1_Diff1__1",
          "Expression:="          , "dB(St(Diff1,Diff1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Differential Pairs"
          ]
        ],
        [
          "NAME:CacheItem",
          "Title:="               , "dB_St_P1_P1__1",
          "Expression:="          , "dB(St(P1,P1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Terminals"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ]
]
    ],
    "CacheSaveKind:="       , "Delta",
    "ConstantDelta:="       , "0s",
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
])
```

Example HFSS Hybrid and Arrays with Beta Parallel Component Mesh Adapt

```
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup2",
    "SolveType:="           , "Single",
    "Frequency:="           , "5GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , False,
    "MaximumPasses:="       , 6,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "ComponentAdaptOption:=", "Fully Independent",
    "PerformFullDesignSolveAtLastPass:=", True,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , "",
    "MaxPass:="             , 10,
    "MinPass:="             , 1,
    "MinConvPass:="         , 1,
    "PerError:="            , 1,
    "PerRefine:="           , 30
  ])
```

InsertSetup [Parametric]

Inserts a new parametric setup.

UI Access	Right-click the **Optimetrics** folder in the project tree, and then click **Add> Parametric** on the shortcut menu.
Parameters	

Name	Type	Description
<Parametric Params>	Array	["NAME:<SetupName>", "SaveFields:=",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<SaveField>, <StartingPoint>, "Sim. Setups:=", <SimSetups>, <SweepDefs>, <SweepOps>, ["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>), ... ["NAME:Goal", <OptiGoalSpec>]]
	<SetupName>	String	Name of the parametric setup.
	<SimSetups>	Array of Strings	An array of Twin Builder solution setup names.
	<SweepDefs>	Array	["NAME:Sweeps", ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>), ... ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>]]
	<SweepData>	String	"<SweepType>, <StartV>, <StopV>, <StepV>"
	<SweepType>	String	The type of sweep data.
	<SyncNum>	Integer	SweepDatas with the same value are synchronized.
	<SweepOps>		["NAME:Sweep Operations", "<OpType>:=, [<VarValue>, …, <VarValue>], ... <OpType>:=, [<VarValue>, …, <VarValue>]]
	<OpType>	String	The sweep operation type.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	InsertSetup ("OptiParametric", <ParametricParams>)
Python Example	oModule.InsertSetup("OptiParametric", ["NAME:ParametricSetup1", "SaveFields:=", true, ["NAME:StartingPoint"], "Sim. Setups:=", ["Setup1"], ["NAME:Sweeps", ["NAME:SweepDefinition", "Variable:=", "$width", "Data:=", "LIN 12mm 17mm 2.5mm", "OffsetF1:=", false, "Synchronize:=", 0], ["NAME:SweepDefinition", "Variable:=", "$length", "Data:=", "LIN 8mm 12mm 2mm", "OffsetF1:=", false, "Synchronize:=", 0]],

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  [
    "NAME:Sweep Operations"
  ],
  [
    "NAME:Goals",
    [
      "NAME:Goal",
      "Solution:=", "Setup1 : LastAdaptive",
      "Calculation:=", "returnloss",
      "Context:=", ""
      [
        "NAME:Ranges",
        "Range:=",
        [
        "Var:=", "Freq", "Type:=","s"
        "Start:=", "8GHz",
        "Stop:=", "8GHz"
        ]
      ],
      [
        "NAME:Goal",
        "Solution:=", "Setup1 : LastAdaptive"
        "Calculation:=", "reflect",
        "Context:=", "",
        [
          "NAME:Ranges",
          "Range:=",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            [
              "Var:=", "Freq",
              "Type:=","s",
              "Start:=", "8GHz",
              "Stop:=", "8GHz"
            ]
          ]
        ]
      ]
    ]
])
```

InsertSetup [Optimization]

Inserts a new optimization setup.

UI Access	Right-click the **Optimetrics** folder in the project tree, and then click **Add > Optimization** on the shortcut menu.
Parameters	See table below

Name	Type	Description
<OptimizationParams>	Array	["NAME:<SetupName>", "SaveFields:=", <SaveField>, <StartingPoint>, "Optimizer:=", <Optimizer>, "MaxIterations:=", <MaxIter>, "PriorPSetup:=", <PriorSetup>, "PreSolvePSetup:=", <Preceed>, <OptimizationVars>, <Constraint>, ["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>, <OptimizationGoalSpec>], ...

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

		["NAME:Goal", <OptiGoalSpec>, <OptimizationGoalSpec>]], "Acceptable_Cost:=", <AcceptableCost>, "Noise:=", <Noise>, "UpdateDesignWhenDone:=", <UpdateDesign>
<OptimizationVars>	Array	["NAME:Variables", "VarName:=", ["i:=",<IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>], ……. "VarName:=", ["i:=", <IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>]]
<MinStepV>	VarValue	The minimum step of the variable.
<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".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<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.

<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>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  Array("NAME:GoalValue", "GoalValueType:=",
  <GoalValueType>,
   "Format:=", <GoalValueFormat>, "bG:=",
  Array("v:=", <GoalValue>)), "Weight:=", <Weight>)
<OptimizationCond>
```

Type: <string>

Either "<=", "==", or ">="

```
<GoalValueType>
```

Type: <string>

Either "Independent" or "Dependent"

```
<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",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"SaveFields:=", false, _
["NAME:StartingPoint", "$length:=", "8mm",
"$width:=", "14.5mm"],
"Optimizer:=", "Quasi Newton",
"MaxIterations:=", 100,
"PriorPSetup:=", "ParametricSetup1",
"PreSolvePSetup:=", true,
["NAME:Variables",
"$length:=", ["i:=", true, "Min:=", "6mm",
"Max:=", "18mm",
"MinStep:=", "0.001mm", "MaxStep:=",
"1.2mm"],
"$width:=", ["i:=", true, "Min:=",
"6.5mm", "Max:=", "19.5mm",
"MinStep:=", "0.001mm", "MaxStep:=",
"1.3mm"]],
["NAME:LCS"],
["NAME:Goals",
["NAME:Goal",
"Solution:=", "Setup1 : LastAdaptive",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Calculation:=", "reflect",
"Context:=", "",
["NAME:Ranges",
"Range:=", ["Var:=", "Freq",
"Type:=", "s",
"Start:=", "8GHz", "Stop:=", "8GHz"]],
"Condition:=", "&lt;=",
["NAME:GoalValue",
"GoalValueType:=", "Independent",
"Format:=", "Real/Imag",
"bG:=", ["v:=", "[0.0001]"]],
"Weight:=", "[1]"]],
"Acceptable_Cost:=", 0.0002,
"Noise:=", 0.0001,
"UpdateDesign:=", true,
"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 [Sensitivity]

Inserts a new sensitivity setup.

<table>
<tr><td>UI Access</td><td colspan="3">Right-click Optimetrics in the project tree, and then click Add>Sensitivity on the shortcut menu.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></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",
<OptiGoalSpec>], ..., ["NAME:Goal",
<OptiGoalSpec>]], "Primary Goal:=". <PrimaryGoalID>,
"PrimaryError:=", <PrimaryError>]</td></tr>
<tr><td><*SensitivityVars*></td><td>Array</td><td>["NAME:Variables",
"VarName:=", Array["i:=", <IncludeVar>,
"Min:=", <MinV>, "Max:=", <MaxV>,
"IDisp:=", <InitialDisp>],...
"VarName:=", ["i:=", <IncludeVar>,
"Min:=", <MinV>, "Max:=", <MaxV>,
"IDisp:=", <InitialDisp>]]</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.

	<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,`
	`"PriorPSetup:=", "",`
	`"PreSolvePSetup:=", true,`
	`["NAME:Variables"],`
	`["NAME:LCS"],`
	`"NAME:Goals",`
	`["NAME:Goal",`
	`"Solution:=", "Setup1 : LastAdaptive",`
	`"Calculation:=", "returnloss",`

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"]]],_
["NAME:Goal",
"Solution:=", "Setup1 : LastAdaptive",
"Calculation:=", "reflect",
"Context:=", "",
["NAME:Ranges",
"Range:=", ["Var:=","Freq",
"Type:=", "s",
"Start:=", "8GHz", "Stop:=", "8GHz"]]],
 "Primary Goal:=", 1,
"PrimaryError:=", 0.001])
```

InsertSetup [Statistical]

Inserts a new statistical setup.

UI Access	Right-click **Optimetrics** in the project tree, and then click **Add** > **Statistical** on the shortcut menu.

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
<*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 is `Uniform`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<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",`
	`  "Context:=", "",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:Ranges", "Range:=", ["Var:=", "Freq", "Type:=", "s", "Start:=", "8GHz", "Stop:=", "8GHz"]]]])

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.

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><OldName></td><td>String</td><td>The name that needs to be replaced</td></tr><tr><td><NewName></td><td>String</td><td>Replacement name</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>Property path</td><td>text string</td><td>Setup property path. See discussion of Property Path</td></tr><tr><td>new Value</td><td>Text String, Number, or Boolean</td><td>New value data type is depending on the property type,</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Python Syntax	SetPropValue(propPath, newValue)
Python Example	`oOptModule.SetPropValue("ParametricSetup1\Enabled", False) //disable Para-`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	metricSetup1 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: *<isBlocking>*; Type: Boolean; Description: An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	SolveAllSetup()
Python Example	oModule.SolveAllSetup()

SolveSetup [Optimetrics]

Solves the specified Optimetrics setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	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

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

DeleteSetups [Optimetrics]

DistrbutedAnalyzeSetup

EditSetup

EnableSetup

ExportDXConfigFile

ExportOptimetricsProfile

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOptimetricsResult

ExportParametricResults

ExportRespSurfaceMinMaxTable

ExportRespSurfaceRefinePoints

ExportRespSurfaceResponsePoints

ExportRespSurfaceVerificationPoints

ExportDOEResponseCurve

ExportDOEResponseCurveSlices

ExportDOEResponseSurface

ExportDOELocalSensitivity

ExportDOELocalSensitivityCurve

GetChildNames [Optimetrics]

GetChildObject [Optimetrics]

GetChildTypes [Optimetrics]

GetName

GetObjPath [Editor]

GetOptimetricResult

GetPropEvaluatedValue

GetPropNames [Optimetrics]

GetPropSIValue

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropValue [Optimetrics]

GetSetupNames [Optimetrics]

GetSetupNamesByType [Optimetrics]

ImportSetup

InsertSetup

PasteSetup [Optimetrics]

RenameSetup [Optimetrics]

SetPropValue [Optimetrics]

SolveSetup [Optimetrics]

SolveAllSetup

CopySetup

Copy the specified Optimetrics setup.

UI Access	NA
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 `true`. If it's `false`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.</td></tr></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	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><th>Name</th><th>Type</th><th>Description</th></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",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"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",
            "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"
            ]
  ],
  "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"],
"UseIterativeSolver:="  , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "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,
        "SkipIERegionSolveDuringAdaptivePasses:=", True
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "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,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "EnableCWRays:="        , False,
                "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"
      ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "ForceSourceToSolve:="  , True,
    "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	See table below
Return Value	None.

Name	Type	Description
<*SetupName*>	String	Name of specified setup.
<*Enable*>	Boolean	Determines whether enable or disable a setup. • **True** - enable setup. • **False** - disable setup.

Ansys 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></table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	[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		

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	Name	Type	Description
	<SetupName>	String	Must be one of the existing Parametric setup names
	<FileName>	String	Must be a valid file path and name with extension of csv, tab, dat or txt
	<bOutputUnits>	Boolean	If set to true, values will be printed with units
Return Value	None		

Python Syntax	ExportParametricResults (<SetupName>, <FileName>, <bOutputUnits>)
Python Example	`oModule.ExportParametricResults(` `"ParametricSetup1", "c:/exportdir/test.csv", False)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportParametricSetupTable

Exports the parametric setup table as a CSV file.

UI Access	Double-click parametric setup. Select **Table** tab. Click **Export**.
Parameters	Name / Type / Description <SetupName> / String / Name of the setup. <filePath> / String / Full path for file export.
Return Value	None

Python Syntax	ExportParametricSetupTable (<SetupName>, <filePath>)
Python Example	`oModule.ExportParametricSetupTable('ParametricSetup1', 'E:/Files/ParametricSetup1_Table.csv')`

ExportRespSurfaceMinMaxTable

Exports min-max table from a response surface to a file

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description *<DOEName>* / String / Name of the Design of Experiments (DOE) setup.

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	Output file name with path.
Return Value	None.		

Python Syntax	ExportRespSurfaceMinMaxTable(*<DOEName>*, *<FileName>*)
Python Example	`oModule.ExportRespSurfaceMinMaxTable("DesignOfExperimentsSetup1", "C:/temp/DesignOfExperimentsSetup1_Min-Max_Search.csv")`

ExportRespSurfaceRefinePoints

Exports refinement points table to a file

UI Access	From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, select **Refinement Points** option under **View**, Click on **Export...**.		
Parameters	Name	Type	Description
	<DOEName>	String	Name of the Design of Experiments (DOE) setup.
	<FileName>	String	Output file name with path.
Return Value	None.		

Python Syntax	ExportRespSurfaceRefinePoints(*<DOEName>*, *<FileName>*)
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")`

ExportDOEResponseCurve

Exports response curve from a response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	

Name	Type	Description
<*DOEName*>	String	Name of the Design of Experiments (DOE) setup.
<*FileName*>	String	Output file name with path.

Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOEResponseCurve("DesignOfExperimentsSetup1", "C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_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.

ExportDOEResponseCurveSlices

Exports response curve slices from a response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOEResponseCurveSlices("DesignOfExperimentsSetup1", "C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOEResponseSurface

Exports response surface to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOEResponseSurface("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

ExportDOELocalSensitivity

Exports local sensitivity to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.		
Parameters	Name	Type	Description
	<*DOEName*>	String	Name of the Design of Experiments (DOE) setup.
	<*FileName*>	String	Output file name with path.
Return Value	None		

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOELocalSensitivity("DesignOfExperimentsSetup1",` `"C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_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.

ExportDOELocalSensitivityCurve

Exports local sensitivity curves to a file.

UI Access	Click on **Export...** From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog.
Parameters	Name / Type / Description <*DOEName*> / String / Name of the Design of Experiments (DOE) setup. <*FileName*> / String / Output file name with path.
Return Value	None

Python Syntax	*DOEName* (<*FileName*>)
Python Example	`oModule.ExportDOELocalSensitivityCurve("DesignOfExperimentsSetup1", "C:/Users/AEDT/R24.1/D/DesignOfExperimentsSetup1_Response_Points.csv")`

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”.

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 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.

Python Syntax	GetChildObject()
Python Example	`oParamSetup = oOptModule.GetChildObject('ParametricSetup1')` `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.

GetChildTypes [Optimetrics]

Use: Gets child types of queried Optimetrics module.

Syntax: GetChildTypes()

Return Value: Array of text string, it can be an empty array if there is no setup is defined. There are six types of setup, they are ['OptiParametric', 'OptiOptimization', 'OptiSensitivity', 'OptiStatistical', 'OptiDesignExplorer', 'OptiDXDOE'].

Python Syntax	GetChildTypes ()
Python Example	`oOptimModule = oDesign.GetChildObject("Optimetrics")` `arrSetupTypes = oOptimModule.GetChildTypes()`

GetName

Returns the design ID and design name of the active design, in that order separated by a semicolon. If the active design is a sub-circuit, it also returns the parent ID and parent design name, listed before the active design information.

To get the *only* the design's name without the parent design name or design IDs, see oDesign.GetDesignName().

UI Access	N/A
Parameters	None.
Return Value	String indicating the name and ID of the active design and parent design (if the active design is a sub-circuit).

Python Syntax	GetName()
Python Example	oDesign.GetName( ) for a subcircuit *EMDesign2* in a parent design *EMDesign1*

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	design_name = oDesign.GetName() 1;EMDesign1/2;EMDesign2 In the returned string: `1` is the parent design ID; `EMDesign1` is the parent design name. `2` is the active design ID; `EMDesign2` is the active design name.

GetObjPath [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()`

GetOptimetricResult

Returns an Optimetric calculation. The specific calculation is determined by the setup.

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	Optimetrics setup name.
	<vars>	Array	Array containing string variable names. Use the **Sweep Definitions** tab in the UI or the <SweepDefs> parameter in the InsertSetup script to determine appropriate inputs.
	<values>	Array	*Optional*. Array containing string values. 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'])`

GetPropEvaluatedValue

Returns the Evaluated-Value for Value-Property and Variable. Returns the Property-value as text string for other property types

> **Note:**
>
> This command is not supported by the EMIT and Circuit design types.

UI Access	N/A		
Parameters	Name	Type	Description

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.
Return Value	String value of the evaluated value.		

Python Syntax	GetPropEvaluatedValue (*<PropName>*)
Python Example	`oVar = oDesign.GetChildObject(" Variables/var")` `oVar.GetPropEvaluatedValue()`

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)`

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.

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	Name: *<PropName>*, Type: String, Description: 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'
	`oCreateBox.GetPropSIValue("xSize")`
	return 0.0004

GetPropValue [Optimetrics]

Returns the property value for a setup property.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	Name | Type | Description property-path | | a child object's property path. See property path discussion here.
Return Value	Returns the value of an setup property.

Python Syntax	GetPropValue(propPath)
Python Example	`oOptModule.GetPropValue("OptimizationSetup1\Optimizer") //get the optimizer name for OptimizationSetup1` `oOptModule.GetPropValue("OptimizationSetup1\Optimizer\Choices") //Get the menu property's menu items. In this case all Optimizer names.`

GetSetupNames [Optimetrics]

Gets a list of Optimetrics setup names

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	IAnsoftCollectionObj – a collection of Optimetrics setup names

Python Syntax	GetSetupNames()
Python Example	`oModule = oDesign.GetModule("Optimetrics")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`setupNames = oModule.GetSetupNames()`

GetSetupNamesByType [Optimetrics]

Gets a list of Optimetrics setup names by type.

UI Access	NA
Parameters	Name | Type | Description <Optimetrics type> | String | Examples: parametric, optimization, statistical, sensitivity
Return Value	Array of Optimetrics setup names of the given type.

Python Syntax	GetSetupNamesByType (<Optimetrics type>)
Python Example	`for name in oModule.GetSetupNamesByType("optimization")` `  AddInfoMessage(str(name))`

ImportSetup

Import an Optimetric setup from a file.

UI Access	NA
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<SetupTypeName>	String	Must be one of "OptiParametric" , "OptiOptimization", "OptiSensitivity", "OptiStatistical", or "OptiDesignExplorer".

	<SetupInfo>	Array	`[NAME:<SetupName>", "FilePath"]`
			`<SetupName>`
			Type: <string>; name of the setup.
			`<FilePath>`
			Type : <string: file path>; must be a valid file path and name.

Return Value	None		

Python Syntax	ImportSetup (<SetupTypeName>, <SetupInfo>)

Python Example	`oModule.ImportSetup ("OptiStatistical",`
	`["NAME:StatisticalSetup1",`
	`"c:/importdir/mySetupInfoFile"])`

InsertSetup

Adds a new solution setup.

UI Access	[product] > **Analysis Setup** > **Add Solution Setup**.		
Parameters	Name	Type	Description
	<*SetupType*>	String	Type of setup to be inserted.
	<*Attributes*>	Array	Structured array. `["NAME:<SetupName>", <NamedParameters>]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<Named Parameters> "AdaptMultipleFreqs:=", <boolean>, "Frequency:=", <string>, "MaxDeltaS:=", <float>, "PortsOnly:=", <boolean>, "UseMatrixConv:=", <boolean>, ### If UseMatrixCov is True, ### ["NAME:Matrix Convergence", <Convergence Array>], MaximumPasses:=", <integer>, "MinimumPasses:=", <integer>, "MinimumConvergedPasses:=", <integer>, "PercentRefinement:=", <integer>, "IsEnabled:=", <boolean>, "BasisOrder:=", <integer>, "DoLambdaRefine:=", <boolean>, "DoMaterialLambda:=", <boolean>, "SetLambdaTarget:=", <boolean>, "Target:=", <float>, "PortAccuracy:=", <integer>, "SaveRadFieldsOnly:=", <boolean>, "SaveAnyFields:=", <boolean>, "ListsForFields:=", <array of string>, "IESolverType:=", <string>, "LambdaTargetForIESolver:=", <float> "UseDefaultLambdaTgtForIESolver:=", <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	InsertSetup(<*SetupType*>, <*Attributes*>)

Python Example	Maxwell 3D AC Magnetic Setup
	`oModule = oDesign.GetModule("AnalysisSetup")`
	`oModule.InsertSetup("AC Magnetic",`
	`[`
	`  "NAME:ACMagnetic_03",`
	`  "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",`
	`  [`
	`  "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"
])
```

HFSS simulation

```
oModule.InsertSetup("HfssDrivenAuto",
["NAME:Setup1",
  "Enabled:=", True,
  "AutoSolverSetting:=", "Balanced",
  ["NAME:Sweeps",
          ["NAME:Sweep",
            "RangeType:=", "LinearStep",
            "RangeStart:=", "1GHz",
            "RangeEnd:=", "10GHz",
            "RangeStep:=", "1GHz"
            ]
  ],
  "SaveRadFieldsOnly:=", False,
  "SaveAnyFields:=", True,
  "Type:=", "Discrete"
  ])

oModule.InsertSetup("HfssDrivenAuto",
        [
                "NAME:Setup2",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "SolveType:="           , "Auto",
                "IsEnabled:="           , True,
        [
                "NAME:MeshLink",
                "ImportMesh:="          , False
        ],
        "AutoSolverSetting:="   , "Balanced",
        [
                "NAME:Sweeps",
                [
                        "NAME:Sweep",
                        "RangeType:="           , "LinearCount",
                        "RangeStart:="          , "0GHz",
                        "RangeEnd:="            , "10GHz",
                        "RangeCount:="          , 501
                ]
        ],
        "SaveRadFieldsOnly:="   , False,
        "SaveAnyFields:="       , True,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "ListsForFields:="      , ["Objectlist2"],
        "Type:="                , "Interpolating"
        ])


oModule.InsertSetup("HfssDriven",
["NAME:Setup3",
        "AdaptMultipleFreqs:=", False,
        "Frequency:=", "5GHz",
        "MaxDeltaS:=", 0.02,
        "PortsOnly:=", False,
        "UseMatrixConv:=", False,
        "MaximumPasses:=", 6,
        "MinimumPasses:=", 1,
        "MinimumConvergedPasses:=", 1,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "PercentRefinement:=", 30,
        "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,
        "SkipIERegionSolveDuringAdaptivePasses:=", True
        "RayDensityPerWavelength:=", 4,
        "MaxNumberOfBounces:="  , 5,
        "InfiniteSphereSetup:=" , "Infinite Sphere1",
        "SkipSBRSolveDuringAdaptivePasses:=", True,
        "PTDUTDSimulationSettings:=", "PTD Correction + UTD Rays",
        "PTDEdgeDensity:="      , 20
                                            ])
```

An HFSS with Hybrid and Arrays setup

```
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("Project1")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "5GHz",
    "MaxDeltaE:="           , 0.1,
    "MaximumPasses:="       , 6,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "RayDensityPerWavelength:=", 4,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "MaxNumberOfBounces:="  , 5,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "EnableSBRSelfCoupling:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    "SkipSBRSolveDuringAdaptivePasses:=", False,
    "UseSBREnhancedRadiatedPowerCalculation:=", True
  ])
```

An HFSS with Hybrid and Arrays Setup for a Component Array with FEBI Preconditioner

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.EditSetup("Fully_Coupled",
  [
    "NAME:Fully_Coupled",
    "SolveType:=" , "Single",
    "Frequency:=" , "1GHz",
    "MaxDeltaS:=" , 0.01,
    "UseMatrixConv:=" , False,
    "MaximumPasses:=" , 10,
    "MinimumPasses:=" , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:=" , 30,
    "ComponentAdaptOption:=", "Fully Coupled",
    "PerformFullDesignSolveAtLastPass:=", True,
    "IsEnabled:=" , True,
    [
      "NAME:MeshLink",
      "ImportMesh:=" , False
    ],
    "BasisOrder:=" , 1,
    "DoLambdaRefine:=" , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "DoMaterialLambda:=" , True,
    "SetLambdaTarget:=" , False,
    "Target:=" , 0.3333,
    "UseMaxTetIncrease:=" , False,
    "PortAccuracy:=" , 2,
    "UseABCOnPort:=" , False,
    "SetPortMinMaxTri:=" , False,
    "DrivenSolverType:=" , "Domain Decomposition",
    "IterativeResidual:=" , 1E-06,
    "DDMSolverResidual:=" , 0.0001,
    "EnhancedLowFreqAccuracy:=", False,
    "EnhancedFEBIPreconditioner:=", True,
    "SaveRadFieldsOnly:=" , True,
    "SaveAnyFields:=" , True,
    "IESolverType:=" , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:=" , "Balanced",
    "InfiniteSphereSetup:=" , "3D",
    "MaxPass:=" , 10,
    "MinPass:=" , 1,
    "MinConvPass:=" , 1,
    "PerError:=" , 1,
    "PerRefine:=" , 30
  ])
```

An SBR+ Setup with Fast Frequency Looping

```
oModule.InsertSetup("HfssDriven",
        [
                "NAME:Setup1",
                "IsEnabled:="           , True,
                [
                        "NAME:MeshLink",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "ImportMesh:="          , False
                ],
                "IsSbrRangeDoppler:="   , False,
                "RayDensityPerWavelength:=", 4,
                "MaxNumberOfBounces:="  , 5,
                "IsMonostaticRCS:="     , True,
                "EnableCWRays:="        , False,
                "RadiationSetup:="      , "",
                "PTDUTDSimulationSettings:=", "None",
                "FastFrequencyLooping:=", True,
                [
                        "NAME:Sweeps",
                        [
                                "NAME:Sweep",
                                "RangeType:="           , "LinearStep",
                                "RangeStart:="          , "1GHz",
                                "RangeEnd:="            , "10GHz",
                                "RangeStep:="           , "1GHz"
                        ]
                ],
                "ComputeFarFields:="    , True
        ])
```

SBR+ Setup with Enhanced Radiated Power Calculation

```
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup2",
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    "IsSbrRangeDoppler:="   , False,
    "RayDensityPerWavelength:=", 4,
    "MaxNumberOfBounces:="  , 5,
    "EnableCWRays:="        , False,
    "RadiationSetup:="      , "Infinite Sphere1",
    "PTDUTDSimulationSettings:=", "None",
    "FastFrequencyLooping:=", False,
    "UseSBRAdvOptionsGOBlockage:=", False,
    "UseSBRAdvOptionsWedges:=", False,
    [
      "NAME:Sweeps",
      [
        "NAME:Sweep",
        "RangeType:="           , "LinearStep",
        "RangeStart:="          , "1GHz",
        "RangeEnd:="            , "10GHz",
        "RangeStep:="           , "1GHz"
      ]
    ],
    "ComputeFarFields:="    , True,
    "UseSBREnhancedRadiatedPowerCalculation:=", True,
    "IsGOBlockageEnabled:=" , False,
    "GOBlockageSurfaceSelfBlock:=", False
  ])
```

Insert 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")
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule.InsertSetup("HfssRFDischarge",
  [
    "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:="  , False,
      "PreservePartnerSoln:=" , False,
      "PathRelativeTo:="      , "SourceProduct",
      "ApplyMeshOp:="         , True
    ],
    [
      "NAME:Excitations",
      [
        "NAME:1:1",
        "Magnitude:="           , "1",
        "Phase:="               , "0deg"
      ],
      [
        "NAME:2:1",
        "Magnitude:="           , "0",
        "Phase:="               , "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:Frequencies",
      "10GHz"
    ],
    "Minimum Power:="       , "0.01",
    "Maximum Power:="       , "1000000",
    "Minimum Pressure:="    , "100pascal",
    "Maximum Pressure:="    , "101325pascal",
    "Postproc Sampling:="   , 500,
    "Temperature:="         , "0cel",
    "BuiltInGas:="          , "Argon"
    "Save Electron Density:=", True,
    "Is Pulsed Signal:="    , True,
    "Duty Cycle:="          , 50
  ])
```

Setup with UseMatrixConv is True and Matrix Convergence Entry Specified as All

```
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      "AllEntries:="          , True,
      "MagLimit:="            , "0.01",
      "PhaseLimit:="          , "2deg",
      "MagMinThreshold:="     , 0.01
    ],
    "MaximumPasses:="       , 12,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "PercentRefinement:="   , 30,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "ACA",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```

Setup with UseMatrixConv as Entries and MeshLink

```
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Frequency:="           , "10GHz",
"MaxDeltaS:="           , 0.02,
"UseMatrixConv:="       , True,
[
  "NAME:Matrix Convergence",
  [
    "NAME:Entries",
    [
         "NAME:Entry",
         "Port1:="          , "A[1,1]P1",
         "Port2:="          , "A[1,1]P1",
         "MagLimit:="               , "0.011",
         "PhaseLimit:="             , "5deg"
    ],
    [
         "NAME:Entry",
         "Port1:="          , "A[4,4]P2",
         "Port2:="          , "A[1,1]P2",
         "MagLimit:="               , "0.041",
         "PhaseLimit:="             , "5deg"
    ]
  ]
],
"MaximumPasses:="       , 1,
"MinimumPasses:="       , 1,
"MinimumConvergedPasses:=", 1,
"PercentRefinement:="   , 30,
"IsEnabled:="           , True,
[
  "NAME:MeshLink",
  "ImportMesh:="          , True,
  "Project:="             , "This Project*",
  "Product:="             , "",
  "Design:="              , "unit",
  "Soln:="                , "Setup1 : LastAdaptive",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
          "NAME:Params",
          "Airbox_dist:="         , "9.9931mm",
          "Scan_Theta:="          , "0deg",
          "Scan_phi:="            , "0deg",
          "VirtualObject_dist:="  , "2.9979mm",
          "coax_inner_rad:="      , "0.125mm",
          "coax_outer_rad:="      , "0.425mm",
          "cut_off:="             , "1.7mm",
          "feedX:="               , "-0.4mm",
          "feedY:="               , "2mm",
          "feed_length:="         , "1mm",
          "patchX:="              , "9.65mm",
          "rot:="                 , "45deg",
          "subH:="                , "0.8mm",
          "subX:="                , "15mm",
          "subY:="                , "15mm"
        ],
        "ForceSourceToSolve:="  , False,
        "PreservePartnerSoln:=" , False,
        "PathRelativeTo:="      , "TargetProject",
        "ApplyMeshOp:="         , True,
        "AdaptPort:="           , False
      ],
      "BasisOrder:="          , 1,
      "DoLambdaRefine:="      , False,
      "DoMaterialLambda:="    , True,
      "SetLambdaTarget:="     , False,
      "Target:="              , 0.3333,
      "UseMaxTetIncrease:="   , False,
      "PortAccuracy:="        , 2,
      "UseABCOnPort:="        , False,
      "SetPortMinMaxTri:="    , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "DrivenSolverType:="    , "Domain Decomposition",
    "IterativeResidual:="   , 0.0001,
    "DDMSolverResidual:="   , 0.0001,
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , ""
  ])
```

Example from design with UseMatrixConv True for AllDiagEntries

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      "AllDiagEntries:="      , True,
      "DiagonalMag:="         , "0.03",
      "DiagonalPhase:="       , "4rad",
      "MagMinThreshold:="     , 0.01,
      "AllOffDiagEntries:="   , True,
      "OffDiagonalMag:="      , "0.05",
      "OffDiagonalPhase:="    , "6deg",
      "MagMinThreshold:="     , 0.01
    ],
```

Example from Modal design with UseMatrixConv True for Entries

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "NAME:Entry",
            "Port1:="          , "Port1",
            "Port2:="          , "Port1",
            "MagLimit:="               , "0.02",
            "PhaseLimit:="             , "5deg"
        ],
        [
            "NAME:Entry",
            "Port1:="          , "Port3",
            "Port2:="          , "Port2",
            "MagLimit:="               , "0.03",
            "PhaseLimit:="             , "5deg"
        ]
      ]
    ],
```

Example from Design with Periodic Ports and UseMaxtrixConv is True

```
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port1",
          "Port2:="               , "Incident_Port1",
          "MagLimit:="            , "0.11",
          "PhaseLimit:="          , "5deg"
        ],
        [
          "NAME:Entry",
          "Port1:="               , "Incident_Port2",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "Port2:="               , "Incident_Port2",
            "MagLimit:="            , "0.22",
            "PhaseLimit:="          , "5deg"
          ]
        ]
      ],
```

Example Terminal Design with UseMatrixConv = True and Expression Cache

```
oProject = oDesktop.SetActiveProject("term-2x2_1_parasolid")
oDesign = oProject.SetActiveDesign("unit")
oModule = oDesign.GetModule("AnalysisSetup")
oModule.InsertSetup("Setup1",
  [
    "NAME:Setup1",
    "SolveType:="           , "Single",
    "Frequency:="           , "10GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , True,
    [
      "NAME:Matrix Convergence",
      [
        "NAME:Entries",
        [
             "NAME:Entry",
             "Port1:="          , "P2",
             "Port2:="          , "P1",
             "MagLimit:="               , "0.05",
             "PhaseLimit:="             , "10deg"
        ]
      ]
    ],
    "MaximumPasses:="       , 15,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "PercentRefinement:="   , 30,
      "IsEnabled:="           , True,
      [
        "NAME:MeshLink",
        "ImportMesh:="          , False
      ],
      "BasisOrder:="          , 1,
      "DoLambdaRefine:="      , True,
      "DoMaterialLambda:="    , True,
      "SetLambdaTarget:="     , False,
      "Target:="              , 0.3333,
      "UseMaxTetIncrease:="   , False,
      "PortAccuracy:="        , 2,
      "UseABCOnPort:="        , False,
      "SetPortMinMaxTri:="    , False,
      "DrivenSolverType:="    , "Iterative Solver",
      "IterativeResidual:="   , 0.0001,
      "DDMSolverResidual:="   , 0.0001,
      "EnhancedLowFreqAccuracy:=", False,
      "SaveRadFieldsOnly:="   , False,
      "SaveAnyFields:="       , True,
      [
        "NAME:ExpressionCache",
        [
          "NAME:CacheItem",
          "Title:="               , "dB_AxialRatioValue_1",
          "Expression:="          , "dB(AxialRatioValue)",
          "Intrinsics:="          , "Phi=\'0deg\' Theta=\'0deg\'",
          "ReportType:="          , "Far Fields",
          [
            "NAME:ExpressionContext",
            "Context:="             , "Infinite Sphere1"
          ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
          "NAME:CacheItem",
          "Title:="               , "dB_St_Diff1_Diff1__1",
          "Expression:="          , "dB(St(Diff1,Diff1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Differential Pairs"
          ]
        ],
        [
          "NAME:CacheItem",
          "Title:="               , "dB_St_P1_P1__1",
          "Expression:="          , "dB(St(P1,P1))",
          "Intrinsics:="          , "",
          "ReportType:="          , "Terminal Solution Data",
          [
            "NAME:ExpressionContext",
            "Diff:="                , "Terminals"
          ]
]
     ],
     "CacheSaveKind:="       , "Delta",
     "ConstantDelta:="       , "0s",
     "IESolverType:="        , "Auto",
     "LambdaTargetForIESolver:=", 0.15,
     "UseDefaultLambdaTgtForIESolver:=", True,
     "IE Solver Accuracy:="  , "Balanced",
     "InfiniteSphereSetup:=" , ""
])
```

Example HFSS Hybrid and Arrays with Beta Parallel Component Mesh Adapt

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule.InsertSetup("HfssDriven",
  [
    "NAME:Setup2",
    "SolveType:="           , "Single",
    "Frequency:="           , "5GHz",
    "MaxDeltaS:="           , 0.02,
    "UseMatrixConv:="       , False,
    "MaximumPasses:="       , 6,
    "MinimumPasses:="       , 1,
    "MinimumConvergedPasses:=", 1,
    "PercentRefinement:="   , 30,
    "ComponentAdaptOption:=", "Fully Independent",
    "PerformFullDesignSolveAtLastPass:=", True,
    "IsEnabled:="           , True,
    [
      "NAME:MeshLink",
      "ImportMesh:="          , False
    ],
    "BasisOrder:="          , 1,
    "DoLambdaRefine:="      , True,
    "DoMaterialLambda:="    , True,
    "SetLambdaTarget:="     , False,
    "Target:="              , 0.3333,
    "UseMaxTetIncrease:="   , False,
    "PortAccuracy:="        , 2,
    "UseABCOnPort:="        , False,
    "SetPortMinMaxTri:="    , False,
    "DrivenSolverType:="    , "Direct Solver",
    "EnhancedLowFreqAccuracy:=", False,
    "SaveRadFieldsOnly:="   , False,
    "SaveAnyFields:="       , True,
    "IESolverType:="        , "Auto",
    "LambdaTargetForIESolver:=", 0.15,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "UseDefaultLambdaTgtForIESolver:=", True,
    "IE Solver Accuracy:="  , "Balanced",
    "InfiniteSphereSetup:=" , "",
    "MaxPass:="             , 10,
    "MinPass:="             , 1,
    "MinConvPass:="         , 1,
    "PerError:="            , 1,
    "PerRefine:="           , 30
  ])
```

PasteSetup [Optimetrics]

Pastes the specified Optimetrics setup.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr></table>
Return Value	None

Python Syntax	PasteSetup (<SetupName>)
Python Example	`oModule.PasteSetup ("OptimizationSetup1")`

RenameSetup [Optimetrics]

Renames the specified Optimetrics setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	**Right-click the setup in the project tree, and then click Rename** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><OldName></td><td>String</td><td>The name that needs to be replaced</td></tr><tr><td><NewName></td><td>String</td><td>Replacement name</td></tr></table>
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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td>Property path</td><td>text string</td><td>Setup property path. See discussion of Property Path</td></tr><tr><td>new Value</td><td>Text String, Number, or Boolean</td><td>New value data type is depending on the property type,</td></tr></table>
Return Value	True if the property is found and the new value is valid. Otherwise return False.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SetPropValue(propPath, newValue)
Python Example	`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: *<isBlocking>*; Type: Boolean; Description: An optional arg that defaults to `True`. If it's `False`, the command immediately returns while the analysis or analyses run in the background. The status of the analyses can be checked with the `AreThereSimulationsRunning` command.
Return Value	None

Python Syntax	SolveAllSetup()
Python Example	`oModule.SolveAllSetup()`

SolveSetup [Optimetrics]

Solves the specified Optimetrics setup.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	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")`

Parametric Script Commands

EditSetup [Parametric]

ExportParametricSetupTable

GenerateVariationData [Parametric]

InsertSetup [Parametric]

EditSetup [Parametric]

Modifies an existing parametric 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 the setup in the project tree, and then click **Properties** on the shortcut menu.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the Setup</td></tr><tr><td><ParametricParams></td><td>List</td><td>List that defines the parameters of the parametric setup; examples are listed below.</td></tr></table>
Return Value	None

Python Syntax	EditSetup (<SetupName>, <ParametricParams>)
Python Example	See EditSetup [Optimization]

ExportParametricSetupTable

Exports the parametric setup table as a CSV file.

UI Access	Double-click parametric setup. Select **Table** tab. Click **Export**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><SetupName></td><td>String</td><td>Name of the setup.</td></tr><tr><td><filePath></td><td>String</td><td>Full path for file export.</td></tr></table>
Return Value	None

Python Syntax	ExportParametricSetupTable (<SetupName>, <filePath>)
Python Example	`oModule.ExportParametricSetupTable('ParametricSetup1', 'E:/Files/Para-`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	metricSetup1_Table.csv')

GenerateVariationData [Parametric]

Generates variation data before parametric solve for CAD integrated project.

UI Access	Right click on the parametric setup in the project tree and select **Generate Variation Data**
Parameters	Name / Type / Description <*SetupName*> / String / Name of the setup to use for the variation data generation
Return Value	None

Python Syntax	GenerateVariationData <*SetupName*>
Python Example	`oModule.GenerateVariationData("ParametricSetup1")`

InsertSetup [Parametric]

Inserts a new parametric setup.

UI Access	Right-click the **Optimetrics** folder in the project tree, and then click **Add> Parametric** on the shortcut menu.		
Parameters	Name	Type	Description
	<Parametric Params>	Array	["NAME:<SetupName>", "SaveFields:=", <SaveField>, <StartingPoint>, "Sim. Setups:=", <SimSetups>, <SweepDefs>, <SweepOps>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>), ... ["NAME:Goal", <OptiGoalSpec>]]
	<SetupName>	String	Name of the parametric setup.
	<SimSetups>	Array of Strings	An array of Twin Builder solution setup names.
	<SweepDefs>	Array	["NAME:Sweeps", ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>), ... ["NAME:SweepDefinition", "Variable:=", <VarName>, "Data:=", <SweepData>, "Synchronize:=", <SyncNum>]]
	<SweepData>	String	"<SweepType>, <StartV>, <StopV>, <StepV>"
	<SweepType>	String	The type of sweep data.
	<SyncNum>	Integer	SweepDatas with the same value are synchronized.
	<SweepOps>		["NAME:Sweep Operations", "<OpType>:=, [<VarValue>, …, <VarValue>], ... <OpType>:=, [<VarValue>, …, <VarValue>]]
	<OpType>	String	The sweep operation type.
Return Value	None		

Python Syntax	InsertSetup ("OptiParametric", <ParametricParams>)
Python Example	oModule.InsertSetup("OptiParametric",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:ParametricSetup1",
  "SaveFields:=", true,
  [
    "NAME:StartingPoint"
  ],
  "Sim. Setups:=", ["Setup1"],
  [
    "NAME:Sweeps",
    [
      "NAME:SweepDefinition",
      "Variable:=", "$width",
      "Data:=", "LIN 12mm 17mm 2.5mm",
      "OffsetF1:=", false,
      "Synchronize:=", 0
    ],
    [
      "NAME:SweepDefinition",
      "Variable:=", "$length",
      "Data:=", "LIN 8mm 12mm 2mm",
      "OffsetF1:=", false,
      "Synchronize:=", 0
    ]
  ],
  [
    "NAME:Sweep Operations"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ],
  [
    "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",
            "Type:=","s",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "Start:=", "8GHz",
                "Stop:=", "8GHz"
            ]
        ]
    ]
  ]
])
```

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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditSetup (<SetupName>, <OptimizationParams>)
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, "MaxNumIteration:=",1000, "MaxSolTimeInSec:=",3600, "RelGradientTolerance:=", 0,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "MinNumIteration:=", 10
  ],
  "CostFuncNormType:=", "L2",
  "PriorPSetup:=", "",
  "PreSolvePSetup:=", True,
  [
    "NAME:Variables"
  ],
  [
    "NAME:LCS"
  ],
  [
    "NAME:Goals",
    [
      "NAME:Goal",
      "ReportType:=", "Standard",
      "Solution:=", "TR",
      [
        "NAME:SimValueContext",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "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"]
    ],
    "Condition:=", "==",
    [
      "NAME:GoalValue",
      "GoalValueType:=", "Independent",
      "Format:=", "Real/Imag",
      "bG:=", ["v:=", "[1;]"]
    ],
    "Weight:=", "[1;]"
  ],
  "Acceptable_Cost:="    , 0,
  "Noise:=", 0.0001,
  "UpdateDesign:=", False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"UpdateIteration:=", 5, "KeepReportAxis:=", True, "UpdateDesignWhenDone:=", True])

InsertSetup [Optimization]

Inserts a new optimization setup.

UI Access	Right-click the **Optimetrics** folder in the project tree, and then click **Add > Optimization** on the shortcut menu.		
Parameters	Name	Type	Description
	<OptimizationParams>	Array	["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>
	<OptimizationVars>	Array	["NAME:Variables", "VarName:=",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			["i:=",<IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>], ……. "VarName:=", ["i:=", <IncludeVar>, "Min:=", <MinV>, "Max:=", <MaxV>, "MinStep:=", <MinStepV>, "MaxStep:=", <MaxStepV>]]
	<MinStepV>	VarValue	The minimum step of the variable.
	<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>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The minimum step of the variable.

<MaxStepV>

Type: <VarValue>

The maximum step of the variable.

<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>
```


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>

Either "<=", "==", or ">="

<GoalValueType>

Type: <string>

Either "Independent" or "Dependent"

<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",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"PreSolvePSetup:=", true,
["NAME:Variables",
"$length:=", ["i:=", true, "Min:=", "6mm",
"Max:=", "18mm",
"MinStep:=", "0.001mm", "MaxStep:=",
"1.2mm"],
"$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",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"Start:=", "8GHz", "Stop:=", "8GHz"]],
"Condition:=", "&lt;=",
["NAME:GoalValue",
"GoalValueType:=", "Independent",
"Format:=", "Real/Imag",
"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.

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 the setup in the project tree, and then click Properties on the shortcut menu</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 Setup</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

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", [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"NAME:AnalysisStopOptions",
"StopForNumIteration:=", True,
"StopForElapsTime:=", False,
"StopForSlowImprovement:=", False,
"StopForGrdTolerance:="        , False,
"MaxNumIteration:=", 1001,
"MaxSolTimeInSec:=", 3600,
"RelGradientTolerance:=", 0,
"MinNumIteration:=", 10
],
"CostFuncNormType:=", "L2",
"PriorPSetup:=", "",
"PreSolvePSetup:=", True,
[
"NAME:Variables"
],
[
"NAME:LCS"
],
[
"NAME:Goals",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
"NAME:Goal",
"ReportType:=", "Standard",
"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",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Format:=", "Real/Imag", "bG:=", ["v:=", "[1;]"]], "Weight:=", "[1;]"]], "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	Name	Type	Description
	<*SensitivityParams*>	Array	["NAME:<SetupName>", "SaveFields:=", <SaveField>, ["Name:StartingPoint"], "MaxIterations:=", <MaxIter>, "PriorPSetup:=", <PriorSetup>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"PreSolvePSetup:=", <Preceed>, <SensitivityVars>, <Constraint>, ["NAME:Goals", ["NAME:Goal", <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",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
["NAME:SensitivitySetup1",
"SaveFields:=", true,
["NAME:StartingPoint"],
"MaxIterations:=", 20,
"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",
```


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"]]]], "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

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	EditSetup (<SetupName>, <StatisticalParams>)
Python Example	`See EditSetup [Optimization]`

InsertSetup [Statistical]

Inserts a new statistical setup.

UI Access	Right-click **Optimetrics** in the project tree, and then click **Add** > **Statistical** on the shortcut menu.

Parameters

Name	Type	Description
<*StatisticalParams*>	Array	`["NAME:<SetupName>", "SaveFields:=", <SaveField>, <StartingPoint>, "MaxIterations:=", <MaxIter>, "PriorPSetup:=", <PriorSetup>, "PreSolvePSetup:=", <Preceed>, <StatisticalVars>, ["NAME:Goals", ["NAME:Goal", <OptiGoalSpec>], …, ["NAME:Goal", <OptiGoalSpec>]]]`
<*StatisticalVars*>	Array	Parameter of <*StatisticalParams*>: `["NAME:Variables", "VarName:=", ["i:=", <IncludeVar>, "Dist:=", <DistType>, "Tol:=", <Tolerance>, "StdD:=", <StdD>, "Min:=", <MinCutoff>, "Max:=", <MaxCutoff>, ...] "VarName:=", ["i:=", <IncludeVar>, "Dist:=", <DistType>, "Tol:=", <Tolerance>, "StdD:=",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			<StdD>, "Min:=", <MinCutoff>, "Max:=", <MaxCutoff>]]
	<DistType>	String	Parameter of <*StatisticalVars*>; distribution can be "Gaussian" or "Uniform".
	<Tolerance>	VarValue	Parameter of <*StatisticalVars*>; the tolerance for the variable when distribution is Uniform
	<StdD>	VarValue	Parameter of <*StatisticalVars*>; the standard deviation for the variable when distribution is Gaussian.
	<MinCutoff>	Double	Parameter of <*StatisticalVars*>; the minimum cut-off for the variable when distribution is Gaussian.
	<MaxCutoff>	Double	Parameter of <*StatisticalVars*>; the maximum cut-off for the variable when distribution is Gaussian.
Return Value	None		

Python Syntax	InsertSetup ("OptiStatistical", <*StatisticalParams*>)
Python Example	oModule.InsertSetup("OptiStatistical", ["NAME:StatisticalSetup1", "SaveFields:=", true, ["NAME:StartingPoint"], "MaxIterations:=", 50, "PriorPSetup:=", "", ["NAME:Variables"], ["NAME:Goals", ["NAME:Goal", "Solution:=", "Setup1 : LastAdaptive", "Calculation:=", "returnloss", "Context:=", "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:Ranges", "Range:=", ["Var:=", "Freq", "Type:=", "s", "Start:=", "8GHz", "Stop:=", "8GHz"]]], ["NAME:Goal", "Solution:=", "Setup1 : LastAdaptive", "Calculation:=", "reflect", "Context:=", "", ["NAME:Ranges", "Range:=", ["Var:=", "Freq", "Type:=", "s", "Start:=", "8GHz", "Stop:=", "8GHz"]]]])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

12 - 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	Name / Type / Description <*name*> / String / Name of the document to be saved. <*saveTo*> / String / File path to save the document to.
Return Value	none

Python Syntax	SaveHtmlDocumentAs(<*name*>, <*saveTo*>)
Python Example	`oModule.SaveHtmlDocumentAs("Design Summary 1", "DS1.html")`

SavePdfDocumentAs

Saves a pre-existing PDF file to a different name and/or location.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Save As** > **PDF**.
Parameters	Name / Type / Description <*name*> / String / Name of the document to be saved. <*saveTo*> / String / File path to save the document at.
Return Value	None.

Python Syntax	SavePdfDocumentAs(<*name*>, <*saveTo*>)
Python Example	`oModule.SavePdfDocumentAs("Design Summary 1", "DS1.pdf")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateAllDocuments

Refreshes the contents of all created documents. This action is made on the folder rather than the individual document.

UI Access	Right click on **Results** > **Documents** in the Project Manager and click **Update All Documents**.
Parameters	None.
Return Value	None.

Python Syntax	UpdateAllDocuments()
Python Example	`oModule.UpdateAllDocuments()`

UpdateDocument

Refreshes the contents of the selected document.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **Update Document**.
Parameters	Name | Type | Description *<name>* | String | Name of the document to be updated
Return Value	None.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	UpdateDocument(<*name*>)
Python Example	`oModule.UpdateDocument("Test UDD Report")`

ViewHtmlDocument

Displays a pre-existing document as HTML.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **View Xml Document**.
Parameters	Name | Type | Description <*name*> | String | Name of the document to be viewed as a HTML
Return Value	None

Python Syntax	ViewHtmlDocument(<*name*>)
Python Example	`oModule.ViewHtmlDocument("Design Summary 1")`

ViewPdfDocument

Displays a pre-existing document as a PDF file.

UI Access	Right click on the created document in the Project Manager under **Results** > **Documents** and click **View PDF Document**.
Parameters	Name | Type | Description <*name*> | String | Name of the document to be viewed as a PDF

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	None.

Python Syntax	ViewPdfDocument(<*name*>)
Python Example	`oModule.ViewPdfDocument("Design Summary 1")`

Example Python Script: Defining a Document

This script creates a user-defined solution and a document.

```
import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("BJT_Inverter")
oDesign = oProject.SetActiveDesign("Nexxim1")
oModule = oDesign.GetModule("UserDefinedSolutionModule")

oModule.CreateUserDefinedSolution("UDS Distance Trace Arithmetic Result1", "SysLib", "TraceArith-
metic/Distance Sweep Trace Arithmetic",
       [
              "Offset 1:=", "0",
              "Scale 1:=", "1",
              "Offset 2:=", "0",
              "Scale 2:=", "1",
              "Operation:=", "Add"
       ],
       [
              [
                     "Standard",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                        "probe1",
                        "Transient",
                        [
                                style="font-family: monospace;">"NAME:Context",
                                style="font-family: monospace;">"SimValueContext:=", [1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"500000000","DT",False,"0.001","NUMLEVELS",False,"0",-
"WE",False,"10us","WM",False,"10us","WN",False,"0ns","WS",False,"0ns"]
                        ],
                        [
                                "Time:=", ["All"]
                        ],
                        [
                                "Probe Component:=", ["V(Port1)"]
                        ],
                        []
                ],
                [
                        "Standard",
                        "probe2",
                        "Transient",
                        [
                                "NAME:Context",
                                "SimValueContext:=", [1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"500000000","DT",False,"0.001","NUMLEVELS",False,"0",-
"WE",False,"10us","WM",False,"10us","WN",False,"0ns","WS",False,"0ns"]
                        ],
                        [
                                "Time:=", ["All"]
                        ],
                        [
                                "Probe Component:=", ["V(Port1)"]
                        ],
                        []
                ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [])
oModule = oDesign.GetModule("UserDefinedDocuments")
oModule.AddDocument(
        [
                "NAME:Test Report",
                "Test Report",
                "SysLib",
                "TestUDDInputs",
                [
                        "NAME:Inputs",
                        [
                                "NAME:UDS1",
                                "Solution",
                                "UDS Distance Trace Arithmetic Result1",
                                1000000,
                                0
                        ],
                        [
                                "NAME:UDS2",
                                "Solution",
                                "UDS Distance Trace Arithmetic Result2",
                                1000000,
                                2
                        ]
                ]
        ],
        [
                "NAME:DocTraces",
                [
                        "NAME:UDS1",
                        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                                "User Defined",
                                "",
                                "UDS Distance Trace Arithmetic Result1",
                                [
                                        "Context:=", ""
                                ],
                                [
                                        "Distance:=", ["All"]
                                ],
                                [
                                        "Probe Component:=", [""]
                                ],
                                []
                        ]
                ],
                [
                        "NAME:UDS2",
                        [
                                "User Defined",
                                "",
                                "UDS Distance Trace Arithmetic Result1",
                                [
                                        "Context:=", ""
                                ],
                                [
                                        "Distance:=", ["All"]
                                ],
                                [
                                        "Probe Component:=", [""]
                                ],
                        []
                ]
        ]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

13 - 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.

	<table><tr><td></td><td></td><td>cified 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 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	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*SoluNames*></td><td>Array</td><td>Array of strings containing names of User Defined Solutions to be deleted.</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	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	See table below
Return Value	String name of update user defined solution.

Name	Type	Description
<*SoluName*>	String	Name of user defined solution to be edited.
<*NewSoluName*>	String	New name for the specified user defined solution.
<*LibType*>	String	Indicates the library where the UDS plugin file is located. This parameter must be one of the following values: "SysLib", "UserLib", "PersonalLib".
<*RelativePath*>	String	The path of the UDS plugin file relative to the "UserDefinedOutputs" sub-directory of the library specified by <*LibType*>.

<*PropList*>	Array	Strings specify name-value pairs corresponding to the UDS properties specified in the plugin file.
		`For example:`
		`["multiply_factor:=", "2.0", "component_name:=", "resistor1"]`

<*ProbeSelections*>	Array	Name of the probe being specified. Note: this must match a probe name specified in the UDS plugin file.
<DynamicProbes>	Array	Array of <ProbeSelection>'s, representing the probes that are used by dynamic-probes.

Ansys 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	Name | Type | Description <*SoluName*> | String | Name of a specified user defined solution.
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.

14 - 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.

UI Access	From the **Network Data Explorer** tab, select **Transforms** in the **NDE** ribbon. Then select **Combine** from the drop-down menu.
Parameters	<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>
Return Value	Network IDispatch. **Note:** Combine returns **None** if the network data does not have the same number of ports.

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	`oModule.ExportNetworkData("", ["Setup1:Sweep1"], 3, "C://Documents/package_HFSSDesign1.s4p",` `['all'], True, 50, "S", -1, 0, 15, True, True, True`

Warning: The following command is preliminary. Its syntax may change. Be prepared to make changes to legacy scripts.

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

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>Data Table (spreadsheet) (*.tab) 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.
Note: The default value is an empty array.</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.

	<renormalize>	Boolean	If **True**, will renormalize the data to the value of renormImpedance before writing. **Note:** The default value is **False**.
	<renormImpedance>	Double	Data will be renormalized to this impedance before writing only if renormalize is **True**. **Note:** The default value is **50 ohms**.
Return Value	Boolean True if file was successfully written; else False.		

Python Syntax	ExportSpreadsheet()

Python Example	`oNDE = oDesktop.GetTool("ndExplorer")`
	`oData = oNDE.Open(<model path>)`
	`success = oData.ExportSpreadsheet("D:\folder\ne133.tab", 0, 1, 12, [], True, 23)`

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	See below.

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.

15 - CompInstance Script Commands

CompInstance commands should be executed by the **oDesign2** or **oPropHost2** object.

```
oDesign2 = CompInstance.GetParentDesign
```

```
oPropHost2 = CompInstance.GetPropHost
```

Callback Scripting Using CompInstance Object

Callback scripts are scripts that can be set in the Property Dialog for individual properties by clicking the button in the Callback column and choosing a script that is saved with the project. Callback scripts can contain any legal script commands including general Ansys script function calls (e.g., GetApplicationName()). In addition, Callback scripts can also call functions on a special object named CompInstance.

You can obtain an interface to a CompInstance in a schematic or layout by calling oEditor.GetCompInstanceFromRefDes(refDes). For more information see Layout Scripting and Schematic Scripting. This interface is also available as a CompInstance object in CompInstance event callbacks, such as placing a component in a layout or schematic.

Definitions

<**propName**> = text string

<**value**> = double

<**valueText**> = text string

<**fileName**> = full path file name

<**choices**> = string containing menu choices separated by commas

<**initialChoice**> = string containing initial choice for menu; must be one of the <choices>

<**scriptName**> = string containing name of script stored in project

<**bool**> is 1 for true or 0 for false

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<**editorName**> is either "Layout" or "SchematicEditor"

The topics for this section include:

CompInstance Functions

CompInstance Functions

Following are commands that can be used to manipulate properties from a CompInstance script.

The topics for this section include:

GetComponentName

GetEditor

GetInstanceID

GetInstanceName

GetParentDesign

GetPropHost

GetPropServerName

GetComponentName

Returns the name of the component corresponding to this CompInstance.

UI Access	NA		
Parameters	Name	Type	Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	None		
Return Value	String name of component (e.g. MS_TRL) and stores it in "name"		

Python Syntax	GetComponentName()
Python Example	`name = CompInstance.GetComponentName()`

GetInstanceID [Component Instance]

Use: Returns the instanceID of the CompInstance.

Command: None

Syntax: GetInstanceID()

Return Value: String

Python Syntax	GetInstanceID()
Python Example	`id = CompInstance.GetInstanceID()`

GetInstanceName [Component Instance]

Use: Returns the instance name of the component corresponding to this CompInstance.

Command: None

Syntax: GetInstanceName()

Return Value: String

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Returns instanceName (e.g. A7) of compInstance and stores it in "name".

Note that the Instance Name is not the same as the RefDes.

GetParentDesign

Use: Returns an interface to the compInstance's parent design. This interface can be used to call Design functions. See: Design Object Script Commands.

Command: None

Syntax: GetParentDesign()

Return Value: Returns interface to design.

Python Syntax	GetParentDesign()
Python Example	`oDesign2 = CompInstance.GetParentDesign()`

GetPropHost

Use: Returns an interface to the PropHost of the CompInstance, which gives access to its properties.This interface can be used to call PropHost functions; for more information see Callback Scripting Using PropHost Object.

Command: None

Syntax: GetPropHost()

Return Value: Returns interface to PropHost.

Python Syntax	GetPropHost()
Python Example	`oPropHost2 = CompInstance.GetPropHost()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropServerName

Returns the PropServerName of the Component corresponding to this CompInstance.

UI Access	Not available
Parameters	None
Return Value	PropServerName in the form of a string

Python Syntax	GetPropServerName()
Python Example	`name = CompInstance.GetPropServerName()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

This page intentionally
left blank.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential
information of ANSYS, Inc. and its subsidiaries and affiliates.

16 - Schematic Scripting

The Schematic scripting interface is a set of commands that match the data changing methods available in the UI of the Schematic, plus some selection and query methods. Examples of the commands available through the scripting interface are adding items, removing items, and modifying items on the Schematic. Identifying objects on the Schematic is done by a unique ID that is generated and returned by all methods that add objects, and by the FindElements and GetSelections methods. This ID can then be passed into commands to modify or remove Schematic objects.

- Since most Schematic objects can have sub-components (such as property displays or segments on a wire, etc), IDs can also be in the format of "TopLevelID:SubComponentIndex".
- The SubComponentIndex is a one-based index into the sub components of the item thus making the second segment of component CAP1 identified by "CAP1:2".
- A SubComponentIndex of zero will refer to the TopLevel item only.
- When an ID is mentioned later in this document, it will refer to either a simple string ID representing a top-level Schematic item or a "TopLevelID:SubComponentIndex" pair unless otherwise stated.
- If a specified ID or SubComponentIndex does not exist, then the function will ignore that entry and proceed with other specified IDs. Even if no valid IDs are specified, the functions will still return success.

The topics for this section include:

- Method Format
- Editor Scripting IDs
- Create Method List
- General Method List
- Property Method List
- Information Method List

Method Format

In the following formats, [Opt=x] appearing after a parameter indicates that the parameter is optional and the default value is x.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

When calling object-creation methods as functions, parenthesis are required in order to retrieve the name of the created object.

All methods to create Schematic objects have the following form:

```
Create[type](VARIANT parameters,
// Array of type specific parameters
VARIANT attributes,
// attributes is in the format of:
["NAME:Attributes",
"Page:=", int, // [Opt=1] Page number (one-based)
// Note: Page 1 always exists
"X:=", double, // X position of the object
"Y:=", double, // Y position of the object
"Angle:=", double, // Rotation angle (radians)
"Flip:=", bool) // True if mirrored
[out,retval] string id]
```

The algorithm used in the create methods is:

1) Create SchAdd*[type]*Command object with parameters passed in

2) Execute the command

All methods to modify Schematic objects have the following form:

```
[modification](Array("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.

```
"Page:=", Page number [optional: Default=1]
"Selections:=", IDs to modify see format below),
VARIANT params
// Array of modification specific parameters)
```

The algorithms in these types of methods will be:

1) Go through ids and get the SelectableObjs they correspond to
2) Select the SelectableObjs found in (1)
3) Select any sub components specified
4) Create the Sch*[modification]*Command
5) Execute the command

ID Format

When IDs are passed to a function it can be in one of the following formats:

- Array of integers of top-level items only
- String of IDs separated by spaces or commas
- Array of strings with each string element being an ID

Point Format

When a method takes an array of points, each element in the array is a string in the format of: "x y", "x,y", or "(x,y)".

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Editor Scripting IDs

Objects in the schematic are identified by text IDs. These IDs are used in scripts to perform actions on objects. These IDs are returned by all methods that add objects, and by the FindElements and GetSelections methods. The IDs are then passed into commands to modify or remove Schematic objects.

Format of IDs for different schematic objects:

Components: CompInst@<CompName>;<compInstID>;<optional schematicID>

Geometric primitive: SchObj@<schematicID>

Global Port/ground :: GPort@<portName>;<schematicID>

Interface Port :: IPort@<portName>;<schematicID>

Page Port: PagePort@<portName>;<schematicID>

Wire : Wire@<netName>;<schematicID>;<segment index list>

Find this information in the Properties Window for the selected object as follows:

<CompName> General tab/CompName

<compInstID> General tab/ID

<schematicID> Symbol tab/SchematicID

<portName> Param Values tab/PortName

<netName> General tab/NetName

<segment index list> numbers separated by commas; Symbol tab/Segmentn - e.g. Segment0 refers to index "0"

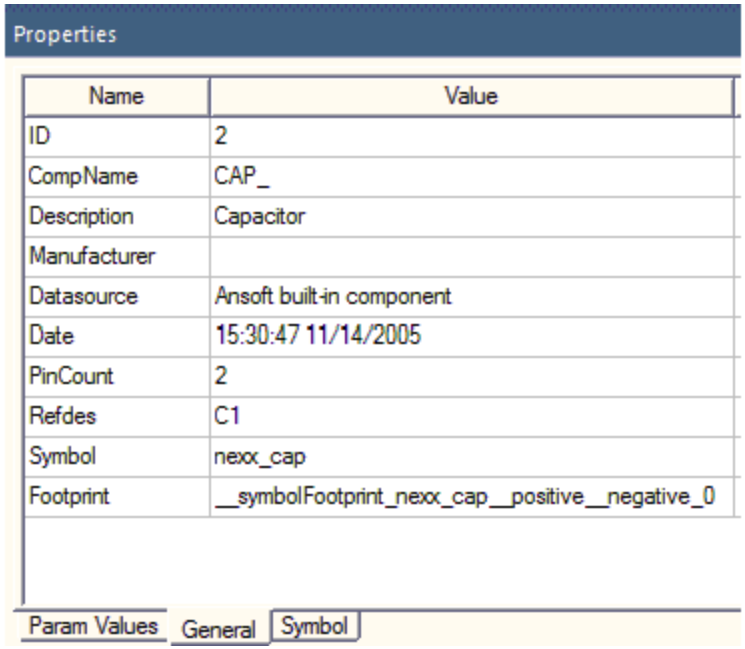
In the Layout, the ID for a selected object is always the Name or Net property in the Footprint tab of the Properties Window

Format for Components

```
["CompInst@CAP_;2;4"]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The format for Components is CompInst@<CompName>;<compInstID>;<optional schematicID>. In the documented example, <CompName> stands for the component name CAP_, <compInstID> stands for its ID 2, and <optional schematicID> is the schematic ID 4. If you select the component, the **Properties** window gets updated and displays its details. For example, the selected object's component name and ID appear in the **General** tab of the **Properties** window as shown below.



The SchematicID is shown in the **Symbol** tab.

Create Method List

This section lists the scripts that are available in the Schematic scripting interface to create and change data.

Script Position Parameters

In Create Method scripts, X and Y position parameters are expressed in meters.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

- If your layout grid units are metric, you may enter the X and Y positions directly into a script as parameters.
- If your layout grid is expressed in mils, you must first convert the component's positional coordinates to meters.

To convert an X or Y position to meters based on the minor grid size:

X or Y position in meters = desired_position x (0.00254 / minor_grid_size)

where desired_position and minor_grid_size have the same units. For example, to position an object at 500mm with a minor grid size of 20mm:

X or Y position in meters = 500 x (0.00254 / 20)

This section lists the following commands:

CreateArc (Schematic Editor)

CreateCircle (Schematic Editor)

CreateComponent (Schematic Editor)

CreateCurve (Schematic Editor)

CreateGlobalPort (Schematic Editor)

CreateGround (Schematic Editor)

CreateLine (Schematic Editor)

CreateNPort (Schematic Editor)

CreatePagePort (Schematic Editor)

CreateIPort (Schematic Editor)

CreatePolygon (Schematic Editor)

CreateRectangle (Schematic Editor)

CreateText (Schematic Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateWire (Schematic Editor)

CreateArc (Schematic Editor)

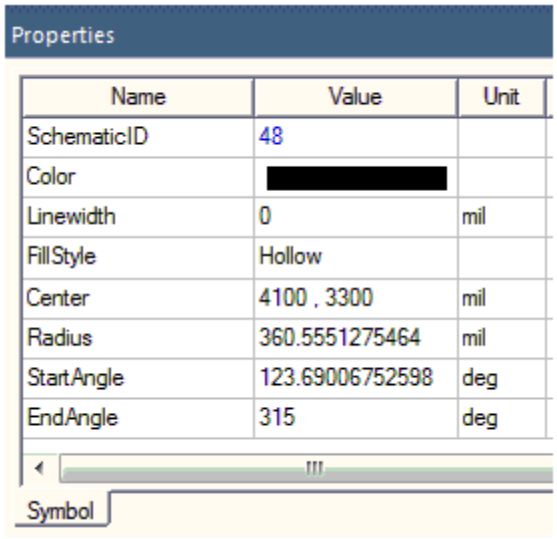
Create an arc

<table>
<tr><td>UI Access</td><td colspan="3">Draw > Primitive > Arc</td></tr>
<tr><td rowspan="3">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><ArcData></td><td>Array</td><td>["NAME:ArcData" _
"x:=", double, _ // X position of the object
"y:=", double, _ // Y position of the object
"Radius:=", double, _ // Radius of the circle
"StartAng:=", double,_ // Start angle of the arc (radians)
"EndAng:=", double, _ // End angle of the arc (radians)
"LineWidth:=", double, // the width of the line, in meters
"Color:=", int, // the RBG value of the arc color
"Id:=", int], // [Opt=New id] Id for this item</td></tr>
<tr><td><Attributes></td><td>Array</td><td>["NAME:Attributes", _
"Page:=", int) _ // [Opt=1] Page number (one-based)
[out,retval] string id]</td></tr>
<tr><td>Return Value</td><td colspan="3">String
Unique id</td></tr>
</table>

Arc is created with the format SchObj@<schematicID>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

The Schematic ID can be found on the **Symbols** tab of the **Properties** window when you select the arc.



Python Syntax	CreateArc(<*ArcData*>, <*Attributes*>)

Python Example	`oEditor.CreateArc(`
	`["NAME:ArcData",`
	`"X:=", -0.004318, "Y:=", -0.00127,`
	`"Radius:=", 0.00297299377732279,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"StartAng:=", 1.9195673303788, "EndAng:=", 3.32144615338227, "Id:=", 10], ["NAME:Attributes", "Page:=", 1])

CreateCircle (Schematic Editor)

Creates a circle.

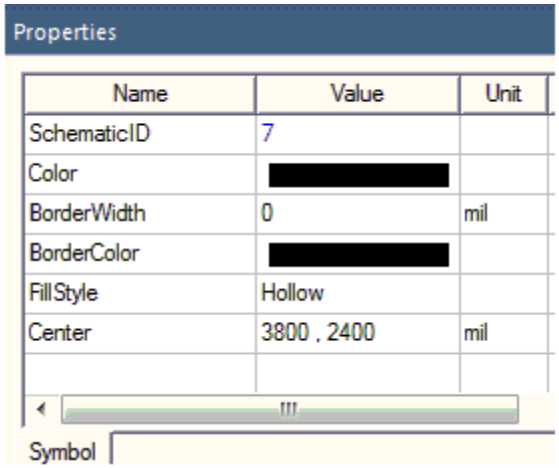
UI Access	**Draw** > **Primitive** > **Circle**		
Parameters	Name	Type	Description
	<*CircleData*>	Array	["NAME:CircleData" "x:=", double, // X position of the object "y:=", double, // Y position of the object "Radius:=", double, // Radius of the circle "LineWidth:=", double, // the width of the circle border, in meters "Bordercolor:=", int, // the RBG value of the border color "Fill:=", int, // the fill pattern id, 0 = hollow, 1 = solid, 2 = NEDiagonal, 3 = OrthoCross, 4 = DiagCross, 5 = NWDiagonal, 6 = Horizontal, 7 = Vertical "Color:=", int, // the RBG value of the circle fill color "Id:=", int], // [Opt=New id] Id for this item
	<*Attributes*>	Array	["NAME:Attributes", "Page:=", int] // [Opt=1] Page number (one-based)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	[out,retval] string id)
Return Value	String Unique id

Circle is created and the returned value has the format SchObj@<schematicID>

The schematic ID of the circle can be located in the **Properties** window on the **Symbols** tab.



Python Syntax	CreateCircle(<*CircleData*>,<*Attributes*>)

Python Example	`oEditor.CreateCircle(`
	`  ["NAME:CircleData",`
	`  "X:=", -0.004572, "Y:=", -0.000508,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Radius:=", 0.001778, "Id:=", 12], ["NAME:Attributes", "Page:=", 1])

CreateComponent (Schematic Editor)

Creates a component of a given type.

UI Access	N/A.
Parameters	See table below.

Name	Type	Description
<params>	Array	Structured array. ["NAME:ComponentProps", "Name:=", <String path to component in library>, "Id:=", <String Component ID>]
<attributes>	Array	Structured array. ["NAME:Attributes", "Page:=", <Integer: page number>, **Note:** For the page number, page 1 always exists. "X:=", <Double: X position of object>, "Y:=", <Double: Y position of the object>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"Angle:=", <Double: rotation angle in degrees>, "Flip:=", <boolean: True if mirrored; else False>, "PinNumber:=", <Integer: index of pin in symbol 0 to n-1>] **Note:** PinNumber is optional and is set to -1 by default. PinNumber affects where the symbol is placed relative to the X and Y parameters. If PinNumber is set to -1, the symbol is placed with its center at (X,Y). If it is a pin index, the symbol is placed so that the pin is located at (X,Y). The PinNumber is not recorded.
Return Value	String containing unique ID in the format: `CompInst@<CompName>;<compInstID>;<optional schematicID>` For example, the return value `CompInst@CAP;4;35` indicates that the component name is `CAP`, its ID is `4`, and its Schematic ID is `35`. You can also see these details in the **Properties** window when you select the created component.		

Python Syntax	CreateComponent(<*params*>, <*attributes*>)

Python Example	`oEditor.CreateComponent(`
	`  ["NAME:ComponentProps",`
	`  "Name:=", "Maxwell Circuit Elements\Passive Elements:Res",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Id:=", "1"
  ],
  ["NAME:Attributes",
    "Page:=", 1,
    "X:=", -0.01524,
    "Y:=", 0.00508,
    "Angle:=", 0,
    "Flip:=", false
  ]
)
```

CreateCurve [Schematic]

Creates a Bezier curve.

UI Access	**Draw** > **Primitive** > **Curve**		
Parameters	Name	Type	Description
	<curve data>	Array	["NAME:CurveData", "Points:=", [ ], //Control Points of curve "LineWidth:=", 0, //Linewidth of curve "Color:=", 0, //Color "Id:=", 1]
	<attributes>	Array	["NAME:Attributes", "Page:=", int] // [Opt=1] Page number

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			(one-based)
Return Value	Name	Type	Description
	<id>	String	Unique ID of this object. Use it to perform actions (such as select and move) on object.

Python Syntax	oEditor.CreateCurve([<*CurveData*>], [<*attributes*>])

Python Example	`curveObjId = oEditor.CreateCurve(`
	`  [`
	`  "NAME:CurveData",`
	`  "Points:=" , ["(-0.149860, 0.233680)","(-0.139700, 0.254000)","(-0.121920, 0.238760)","(-0.147320, 0.226060)","(-0.139700, 0.218440)","(-0.121920, 0.220980)"],`
	`  "LineWidth:=" , 0,`
	`  "Color:=" , 0,`
	`  "Id:=" , 7`
	`  ],`
	`  [`
	`  "NAME:Attributes",`
	`  "Page:=" , 1`
	`])

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

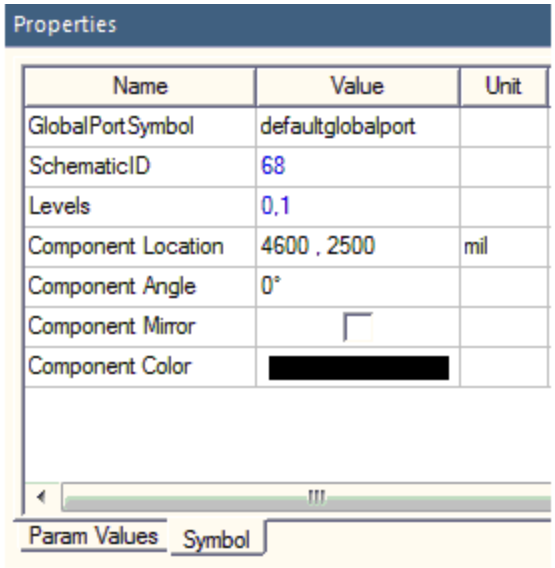
CreateGlobalPort (Schematic Editor)

Creates a global port and enables placement on the schematic.

UI Access	**Draw** > **Global Port**		
Parameters	Name	Type	Description
	<portname>	String	Name for this port - leave empty to use default unique name
	<id>	Integer	ID number for the port - leave empty to use default (recommended)
	<pagenumber>	Integer	The (1-based) page number of the schematic page this port should be added to - leave empty to use default (1)
	<xval>	Double	X position of the object
	<yval>	Double	Y position of the object
	<angle>	Double	Rotation of the object, in radians. Rotations of multiples of 90 degrees are valid. Other values will be adjusted to the nearest valid angle.
	<degrees>	Double	Rotation of the object, in degrees. Rotations of multiples of 90 degrees are valid. Other values will be adjusted to the nearest valid angle
	<flip>	Boolean	True if mirrored
Return Value	String PortID, the identifier that can be used for this port in script operations involving this schematic		

Global port is created and the return value has the format GPort@<portName>;<schematicID>
The Schematic ID can be found in the **Properties** window on the **Symbols** tab when you select the global port.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Python Syntax	CreateGlobalPort(["NAME:GlobalPortProps", "Name:=", <*portname*>, "Id:=", <*id*>], ["NAME:Attributes", "Page:=", <*pagenumber*>, "X:=", <*xval*>, "Y:=", <*yval*>, "Angle:=", <*angle*> or "Degrees:=", <*degrees*> , "Flip:=", <*flip*>])

Python Example	`oEditor.CreateGlobalPort`
	`(["NAME:GlobalPortProps", "Name:=", "G_0", "Id:=", 1],`
	`["NAME:Attributes", "Page:=", 1, "X:=", -0.02286, "Y:=,`
	`0.00254, "Angle:=", 0, "Flip:=", false])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CreateGround (Schematic Editor)

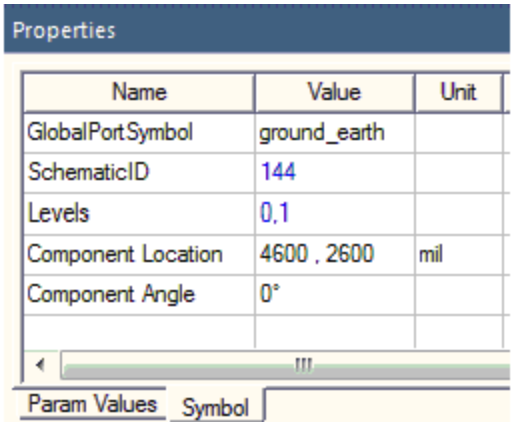
Creates a ground.

<table>
<tr><td>UI Access</td><td colspan="3">Draw > Ground</td></tr>
<tr><td rowspan="3">Parameters</td><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><GroundProps></td><td>Array</td><td>["NAME:GroundProps",
"Id:=", int), // [Opt=New id] Id for this item]</td></tr>
<tr><td><Attributes></td><td>Array</td><td>["NAME:Attributes",
"Page:=", int, _ // [Opt=1] Page number (one-based)
Note: Page 1 always exists
"X:=", double, _ // X position of the object
"Y:=", double, _ // Y position of the object
"Angle:=", double, _ // Rotation angle (radians)
"Flip:=", bool), // True if mirrored
[out,retval] string id)]</td></tr>
<tr><td>Return Value</td><td colspan="3">String
Unique id</td></tr>
</table>

Ground is created and the returned value has the format GPort@<portname>;<schematicID>

The Schematic ID for ground can be found on the **Symbols** tab of the **Properties** window when you select the object.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Python Syntax	CreateGround()

Python Example	`oEditor.CreateGround`
	`(["NAME:GroundProps", "Id:=", 8),`
	`["NAME:Attributes", "Page:=", 1, "X:=",`
	`-0.04064, "Y:=", -0.00254, "Angle:=",`
	`0, "Flip:=", false])`

CreateLine (Schematic Editor)

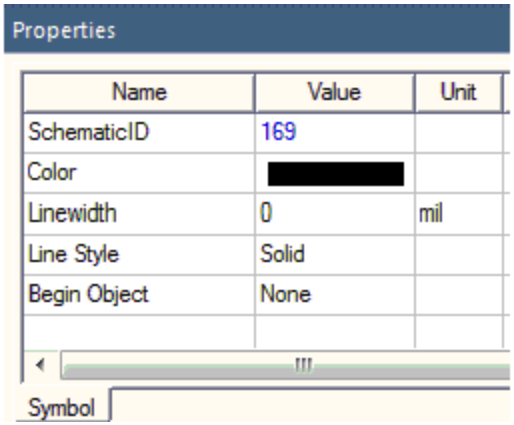
Creates a line.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<table>
<tr><td>UI Access</td><td>Draw > Primitive > Line</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><LineData></td><td>Array</td><td>["NAME:LineData"
"Points:=", Array of points,
"LineWidth:=", double, // the width of the line, in meters
"Color:=", int, // the RBG value of the line color
"Id:=", int), // [Opt=New id] Id for this item</td></tr>
<tr><td><Attributes></td><td>Array</td><td>["NAME:Attributes",
"Page:=", int] // [Opt=1] Page number (one-based)
[out,retval] string id)</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>String
Unique id</td></tr>
</table>

Line is created and the returned value has the format SchObj@<schematicID>

The Schematic ID can be found on the **Symbol** tab of the **Properties** window when you select the line.



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateLine(<*LineData*>,<*Attributes*>)

Python Example	`oEditor.CreateLine (`
	`  ["NAME:LineData",`
	`  "Points:=", ["(-0.055652, 0.020669)",`
	`  "(-0.036923, 0.011257)", "(-0.023144, 0.018049)"],`
	`  "LineWidth:=", 0, "Color:=", 0, "Id:=", 22)],`
	`  ["NAME:Attributes", "Page:=", 1])`

CreateNPort (Schematic Editor)

Use: Creates an N-Port definition and component and adds them to the current project, layout, and schematic.

Command: CreateNPort

Syntax: **CreateNPort**

```
Array("NAME:Contents",
"definition_name:=",<nport_definition_name>,
"placement:=", <component_placement>,
"layer:=", <placement_layer_name>,
<nport_data_definition>)
```

<nport_definition_name>:

quoted string (name of the component definition)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<component_placement>:

```
Array("x:=", <value>, // x coordinate
"y:=", <value>, // y coordinate
"scaling:=", <value>) // double
```

Return Value: Returns the name of the newly created component.

<nport_data_definition**>** - see the I/O format in TODO

CreatePagePort (Schematic Editor)

Creates a PagePort.

UI Access	**Draw** > **Page Connector**		
Parameters	Name	Type	Description
	<PagePortProps>	Array	["NAME:PagePortProps"], ["NAME:Attributes", "Page:=", int, _ // [Opt=1] Page number (one-based) Note: Page 1 always exists "X:=", double, _ // X position of the object "Y:=", double, _ // Y position of the object "Angle:=", double, _ // Rotation angle (radians) "Flip:=", bool), // True if mirrored [out,retval] string id]

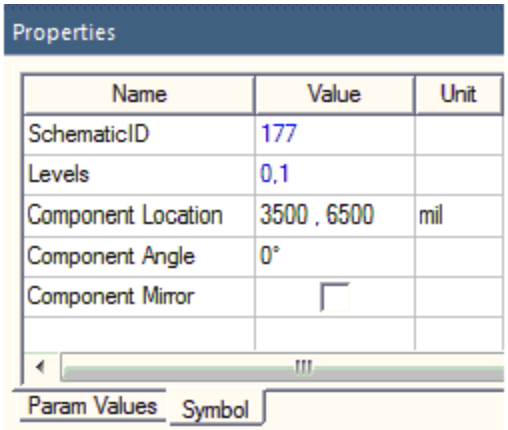
Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value	String Unique id

The pageport is created and it has the following format:

`PagePort@<portName>;<schematicID>`

The schematic ID can be found in the **Properties** window on the **Symbols** tab when you select the pageport.



Python Syntax	CreatePagePort(<PagePortProps>)

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.CreatePagePort`
	`(["NAME:PagePortProps", "Name:=", "pageport_0", "Id:=",`
	`12], ["NAME:Attributes", "Page:=", 1, "X:=", -0.0381, "Y:=", -0.0127, "Angle:=", _`
	`0, "Flip:=", false])`

CreateIPort (Schematic Editor)

Creates an interface port.

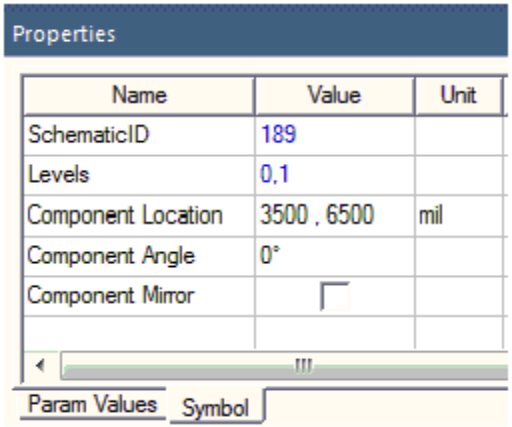
UI Access	**Draw** > **Interface Port**
Parameters	See table below
Return Value	String

Name	Type	Description
<IPortProps>	Array	`["NAME:IPortProps"],` "Name:=", string), // [Opt=default name] Name for this port "Id:=", int), // Port id number
<Attributes>	Array	`["NAME:Attributes",` "Page:=", int] // [Opt=1] Page number (one-based) **Note:** Page 1 always exists "X:=", double, _ // X position of the object "Y:=", double, _ // Y position of the object "Angle:=", double, _ // Rotation angle (radians) "Flip:=", bool), // True if mirrored [out,retval] string id)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Unique id

Interface port is created and the returned value has the following format:
IPort@<portName>;<schematicID>
The interface port schematic ID can be found in the **Properties** window on the **Symbols** tab.



Python Syntax	CreateIPort(<*PortProps*>, <*Attributes*>)
Python Example	`oEditor.CreateIPort (["NAME:IPortProps", "Name:=", "Port1", "Id:=", 4], ["NAME:Attributes","Page:=", 1, "X:=", -0.0381, "Y:=", 0.0127, "Angle:=" 0, "Flip:=", false])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

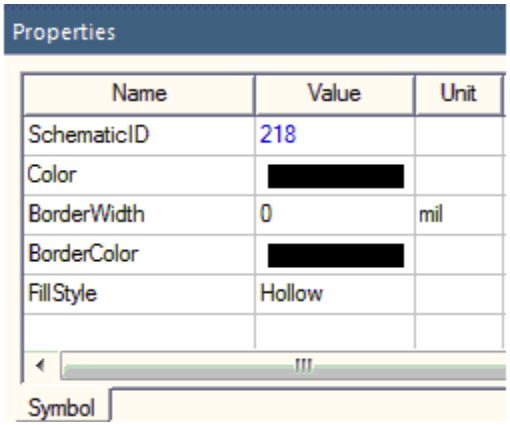
CreatePolygon (Schematic Editor)

Creates a polygon.

<table>
<tr><td>UI Access</td><td colspan="3">Draw > Primitive > Polygon.</td></tr>
<tr><td>Parameters</td><td colspan="3">
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><PolygonData></td><td>Array</td><td>["NAME:PolygonData"
"Points:=", Array of points,
"LineWidth:=", double, // the width of the Polygon border, in meters
"Bordercolor:=", int, // the RBG value of the border color
"Fill:=", int, // the fill pattern id, 0 = hollow, 1 = solid, 2 = NEDiagonal,
3 = OrthoCross, 4 = DiagCross, 5 = NWDiagonal, 6 = Horizontal, 7 = Vertical
"Color:=", int, // the RBG value of the Polygon fill color
"Id:=", int], // [Opt=New id] Id for this item</td></tr>
<tr><td><Attributes></td><td>Array</td><td>["NAME:Attributes",
"Page:=", int] // [Opt=1] Page number (one-based)
[out,retval] string id]</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td colspan="3">String with the format SchObj@<schematicID></td></tr>
</table>

The Schematic ID can be found in the **Properties** window on the **Symbol** tab when you select the polygon.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Python Syntax	CreatePolygon()

Python Example	`oEditor.CreatePolygon(`
	`  ["NAME:PolygonData",`
	`  "Points:=", ["(-0.058951, 0.006308)", "(-0.046142, 0.010480)",`
	`  "(-0.046239, 0.001067)")],`
	`  "LineWidth:=",0,`
	`  "BorderColor:=", 0,`
	`  "Fill:=", 0,`
	`  "Color:=",0,`
	`  "Id:=", 27`

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", "Page:=", 1]])

CreateRectangle (Schematic Editor)

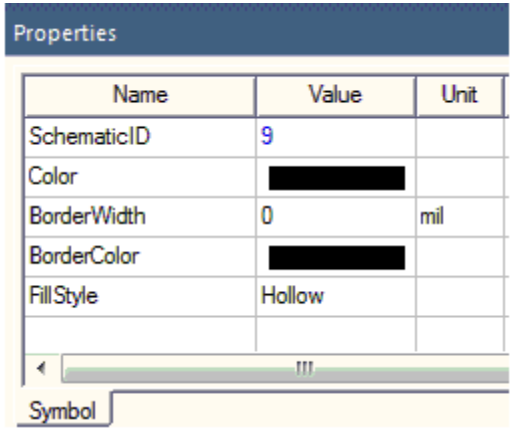
Create a rectangle.

UI Access	**Draw** > **Primitive** > **Rectangle**		
Parameters	Name	Type	Description
	<RectData>	Array	["NAME:RectData" "x1:=", double, // X position of the upper left "y1:=", double, // Y position of the upper left "x2:=", double, // X position of the lower right "y2:=", double", // Y position of the lower right "LineWidth:=", double, // the width of the Rectangle border, in meters "Bordercolor:=", int, // the RBG value of the border color "Fill:=", int, // the fill pattern id, 0 = hollow, 1 = solid, 2 = NEDiagonal, 3 = OrthoCross, 4 = DiagCross, 5 = NWDiagonal, 6 = Horizontal, 7 = Vertical "Color:=", int, // the RBG value of the Rectangle fill color "Id:=", int] // [Opt=New id] Id for this item
	<*Attributes*>	Array	["NAME:Attributes", "Page:=", int] // [Opt=1] Page number (one-based)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	[out,retval] string id)
Return Value	String with the format SchObj@<schematicID>

The Schematic ID can be found in the **Properties** window on the **Symbol** tab when you select the rectangle.



Python Syntax	CreateRectangle(<*RectData*>, <*Attributes*>)

Python Example	`oEditor.CreateRectangle`
	`(["NAME:RectData", "X1:=", -0.0755449856733525, "Y1:=",`
	`0.0335755491881566, "X2:=", -0.0611831900668577, "Y2:=",`
	`0.0254242597898758, "LineWidth:=", 0, "BorderColor:=", 0, "Fill:=", 0, "Color:=", 0,`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Id:=", 31], ["NAME:Attributes", "Page:=", 1])

CreateText (Schematic Editor)

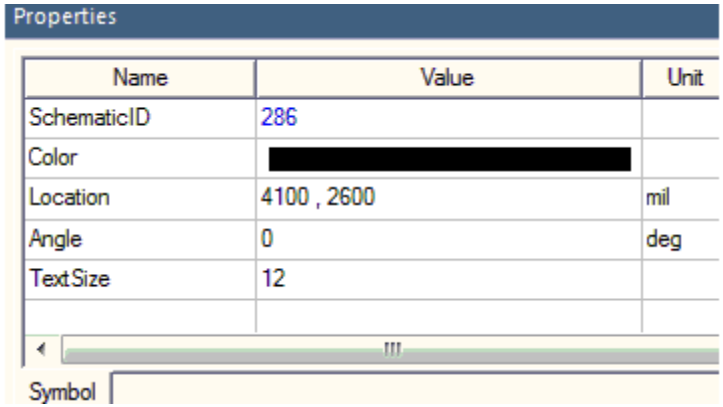
Creates text.

UI Access	**Draw** > **Primitive** > **Text**		
Parameters	Name	Type	Description
	<*TextData*>	Array	["NAME:TextData" "x:=", double, // X position of the object "y:=", double, // Y position of the object "Text:=", string, // Text to display "Color:=", int, // the RBG value of the text color "Id:=", int], // Id for this item "ShowRect:=", bool, // true or false, whether to show the highlight rectangle for the text "X1:=", double, // the text rectangle left X value, in meters "Y1:=", double, // the text rectangle upper Y value, in meters "X2:=", double, // the text rectangle right X value, in meters "Y2:=", double, // the text rectangle lower Y value, in meters "RectLineWidth:=", double, // the width of the rectangle border, in meters "RectBordercolor:=", int, // the RBG value of the rectangle border

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			color "RectFill:=", int, // the rectangle fill pattern id, 0 = hollow, 1 = solid, 2 = NEDiagonal, 3 = OrthoCross, 4 = DiagCross, 5 = NWDiagonal, 6 = Horizontal, 7 = Vertical "RectColor:=", int, // the RBG value of the rectangle fill color
	<Attributes>	Array	["NAME:Attributes", "Page:=", int] // [Opt=1] Page number (one-based) [out,retval] string id)
Return Value	String with the format SchObj@<SchematicID> Unique id		

The Schematic ID can be found in the **Properties** window on the **Symbols** tab when you select the text.



Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	CreateText()
Python Example	oEditor.CreateText(["NAME:TextData", "X:=", -0.0764183381088826, "Y:=", 0.040174212034384, "Size:=", 12, "Angle:=", 0, "Text:=", "Control Circuit", "Color:=", 0, "Id:=", 34, "ShowRect:=", false, "X1:=", -0.0793287547755609, "Y1:=", 0.0407033787010528, "X2:=", -0.0502245881087778,_ "Y2:=", 0.035411712034365, "RectLineWidth:=", _ 0, "RectBorderColor:=", 0, "RectFill:=", 0, "RectColor:=", 0], ["NAME:Attributes", "Page:=", 1])

CreateWire (Schematic Editor)

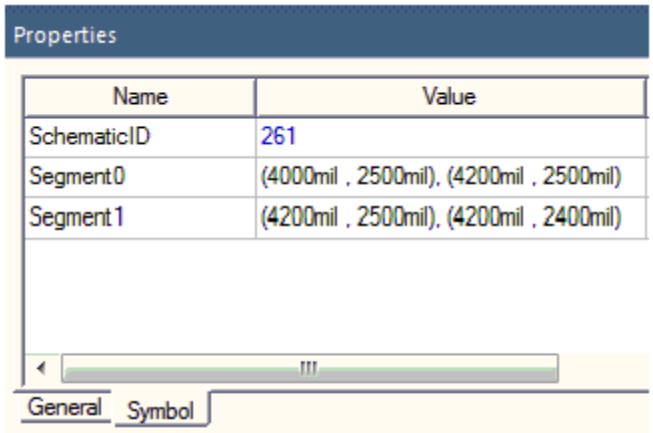
Create a wire.

UI Access	**Draw** > **Wire**		
Parameters	Name	Type	Description
	<*WireData*>	Array	["NAME:WireData" "Name:=", string), // [Opt=default name] Name for this wire "Id:=", int), // Wire idnum "Points:=", Points on wire]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	<Attributes>	Array	["NAME:Attributes", "Page:=", int, _ // [Opt=1] Page number (one-based) Note: Page 1 always exists [out,retval] string id]
Return Value	String with the format Wire@<netName>;<schematicID>;<segment index list>		

The Schematic ID can be found in the **Symbol** tab of the **Properties** window when you select the wire. The wire net name appears in the **General** tab. The parameter segment index list indicates the number of segments, *i.e.* the segment count.



Python Syntax	CreateWire(<*WireData*>, <*Attributes*>)
Python Example	oEditor.CreateWire(["NAME:WireData",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Name:=", "", "Id:=", 41, "Points:=", ["(-0.033020, 0.015240)", "(-0.017780, 0.015240)", "(-0.017780, 0.012700)"]], ["NAME:Attributes", "Page:=", 1])

General Method List

This section presents the general script methods that are available.

The general method script commands are listed here.

Activate (Schematic Editor)

AddPinGrounds (Schematic Editor)

AddPinIPorts (Schematic Editor)

AddPinPageConnectors (Schematic Editor)

AddPinCustomTerminations (Schematic Editor)

AlignHorizontal (Schematic Editor)

AlignVertical (Schematic Editor)

BringToFront (Schematic Editor)

Close Editor (Schematic Editor)

Copy (Schematic Editor)

CopyData [Schematic Editor]

CopySubdesign [Schematic Editor]

CreatePage (Schematic Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Cut (Schematic Editor)

DeactivateOpen (Schematic Editor)

DeactivateShort (Schematic Editor)

Delete (Schematic Editor)

DeletePage (Schematic Editor)

ElectricRuleCheck (Schematic Editor)

ExportImage (Schematic Editor)

FindElements (Schematic Editor)

FitToBorder [Schematic Editor]

GridSetup (Schematic Editor)

FlipHorizontal (Schematic Editor)

FlipVertical (Schematic Editor)

Move (Schematic Editor)

NameNets (Schematic Editor)

PageBorders (Schematic Editor)

Pan (Schematic Editor)

Paste (Schematic Editor)

PasteData [Schematic Editor]

PasteDesign (Schematic Editor)

PushExcitations (Schematic Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Rotate (Schematic Editor)

ShowVariableBlock (Schematic Editor)

SelectAll (Schematic Editor)

SelectPage (Schematic Editor)

SendToBack(Schematic Editor)

SetPageData [Schematic]

SortComponents (Schematic Editor)

ZoomArea (Schematic Editor)

ZoomIn (Schematic Editor)

ZoomOut (Schematic Editor)

ZoomPrevious (Schematic Editor)

ZoomToFit (Schematic Editor)

Wire (Schematic Editor)

Activate (Schematic Editor)

Enables items previously disabled.

UI Access	**Edit** > **Activate**.		
Parameters	Name	Type	Description
	<*Selections*>	Array	List of disabled items to activate.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Activate(["NAME:Selections","Selections:=", [<components>]])

Python Example	`oEditor.Activate(`
	`  [`
	`  "NAME:Selections",`
	`  "Selections:=", ["CompInst@IPROBE;118;129", "CompInst@V_DC;121;163"]`
	`  ])`

AddPinGrounds (Schematic Editor)

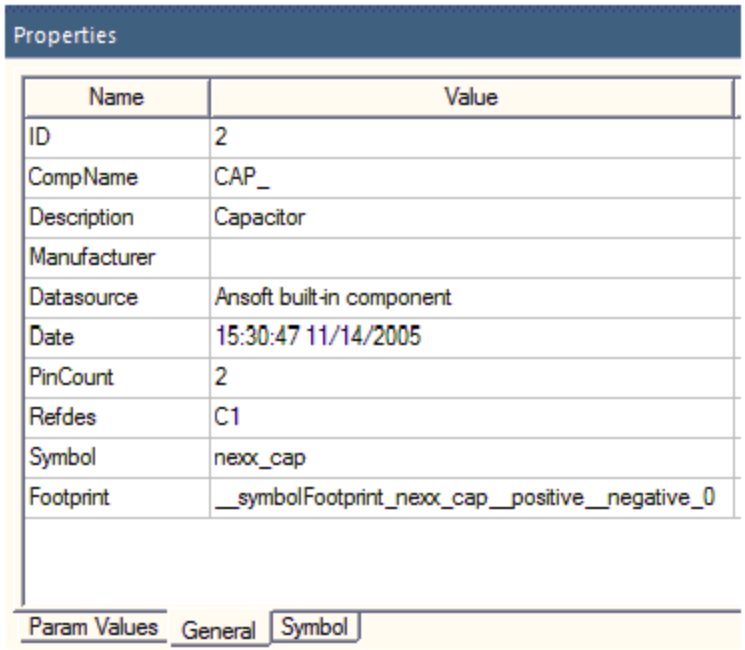
Adds grounds at all unconnected pins.

```
AddPinGrounds (
Array("NAME:Selections", _ // PagePort Name
"Selections:=", _ // Net Name
Array("CompInst@CAP_;2;4"))
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

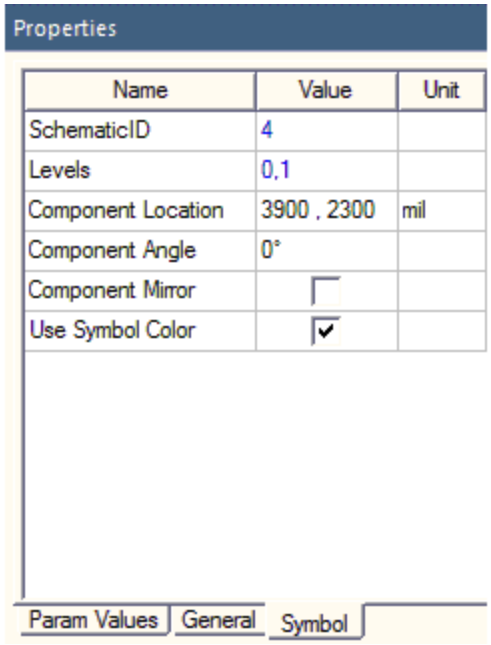
Note:

The format for Components is CompInst@<CompName>;<compInstID>;<optional schematicID>. In the documented example, <CompName> stands for the component name CAP_, <compInstID> stands for its ID 2, and <optional schematicID> is the schematic ID 4. If you select the component, the **Properties** window gets updated and displays its details. For example, the selected object's component name and ID appear in the **General** tab of the **Properties** window as shown below.



To view the SchematicID click the **Symbol** tab.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.



Python Syntax	AddPinPageConnectors (<*Selections*>)
Python Example	`oEditor.AddPinPageconnectors["NAME:Selections", "Selections:=", [ "CompInst@C;1;1"]]`

AddPinlPorts (Schematic Editor)

Adds Interface Ports at unconnected pins of all selected components. Each port has a unique 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	**Draw** > **Add at unconnected pins** > **Interface Ports**
Parameters	Name: <IntPortName,NetName> — Type: Array — Description: `["NAME:Selections", _ // Interface Port Name` "Selections:=", _ // Net Name]
Return Value	None

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	AddPinIPorts ()
Python Example	`oEditor.AddPinIPorts` `["NAME:Selections", "Selections:=", ["CompInst@R;2;37"]]`

AddPinPageConnectors (Schematic Editor)

Adds Page Ports at unconnected pins of all selected components.

UI Access	**Draw** > **Add at unconnected pins** > **Grounds**.
Parameters	Name: *<Name>* — Type: String — Description:
	Name: *<Selections>* — Type: Array — Description: List of components to add page ports to
Return Value	None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	AddPinPageConnectors (<*Name*>,<*Selections*>)

Python Example	`oEditor.AddPinPageConnectors(`
	`  ["NAME:Selections", "Selections:=",`
	`  ["CompInst@C;1;1"]`
	`  ])`

AddPinCustomTerminations (Schematic Editor)

Adds Custom Terminations at unconnected pins of all selected components.

UI Access	Draw > Add at unconnected pins> Custom Terminations.		
Parameters	**Name**	**Type**	**Description**
	ComponentInstanceName	<string>	Instance Name of the unconnected pin that needs to be terminated.
	PinName	<string>	Pin name of the unconnected pin.
	TerminationComponentName	<string>	Optional parameter. Definition Name of the custom component that is being terminated at the unconnected pin.
	TermName	<string>	"Interface Ports" "Page Ports" "Grounds" "<PinName of the termination component>"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	Terminationtype	<INT>	0 (Interface Port), 1 (Page Port), 2 (Ground Port), 3 (Custom Termination)
Return Value	None		

Python Syntax	AddPinCustomTerminations ()

Python Example	`oEditor.AddPinCustomTerminations(`
	`["NAME:Connections",`
	`"Connection1:=", ["InstanceName:=" , "CompInst@etch_sline_s20_z70_1;1;1" , "PinName:=" "Port1" , "TermName:=" , "Interface Ports" , "TermType:=", 1],`
	`"Connection2:=", ["InstanceName:=" , "CompInst@etch_sline_s20_z70_1;1;1" , "PinName:=", "Port3" , "TermName:=" , "Page Ports" , "TermType:=", 2],`
	`"Connection3:=", ["InstanceName:=" , "CompInst@etch_sline_s20_z70_1;1;1" , "PinName:=" "Port7" , "TermComponent:=" , "test" , "TermName:=" , "Port1" , "TermType:=" , 4],`
	`"Connection4:=", ["InstanceName:=" , "CompInst@etch_sline_s20_z70_1;1;1" , "PinName:=" "Port10","TermComponent:=" , "oneport" , "TermName:=" , "Port1" , "TermType:=" , 4]`
	`]`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AlignHorizontal (Schematic Editor)

Aligns items horizontally.

<table>
<tr><td>UI Access</td><td colspan="3">Draw > Align Horizontal</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>["NAME:Selections",
"Page:=", page number, // [Opt=1] Page number
"Selections:=", IDs to modify]</td></tr>
<tr><td><AlignParameters></td><td>Array</td><td>["NAME:AlignParameters", _
"Disconnect:=", bool // [Opt=0] Should wires disconnect
"Rubberband:=", bool] // [Opt=1] Should wires staircase
Note: Alignment occurs relative to the first item in ids</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td colspan="3">Value</td></tr>
</table>

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	AlignHorizontal()

Python Example	`oEditor.AlignHorizontal (`
	`  ["NAME:Selections",`
	`  "Selections:=",["CompInst@R;2;4", "CompInst@C;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.

	], ["NAME:AlignParameters", "Disconnect:=", false, "Rubberband:=", false])

AlignVertical (Schematic Editor)

Aligns items vertically.

UI Access	Draw>Align Vertical		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", IDs to modify]
	<Align Parameters>	Array	["NAME:AlignParameters", "Disconnect:=", bool // [Opt=0] Should wires disconnect "Rubberband:=", bool] // [Opt=1] Should wires staircase Note: Alignment occurs relative to the first item in ids
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	AlignVertical()

Python Example	`oEditor.AlignVertical(`
	`  ["NAME:Selections",`
	`  "Selections:=", ["CompInst@R;2;4", "CompInst@C;1;1"]`
	`  ],`
	`  ["NAME:AlignParameters",`
	`  "Disconnect:=", false,`
	`  "Rubberband:=",`
	`  false`
	`  ])`

BringToFront (Schematic Editor)

Bring the selected object to the front of the view

UI Access	Draw>Bring to Front		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME: Selections", "Selections:=",[<Selected Components>]] <Selected Components>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			CompInst@DefName; ID, schematic ID , CompInst@DefName; ID, schematic ID CompInst@R;1,2
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	BringToFront( )
Python Example	`oEditor.BringToFront( ["NAME:Selections", "Selections:=", ["SchObj@1"]])`

CloseEditor (Schematic Editor)

Close an Editor

UI Access	NA		
Parameters	Name	Type	Description
	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.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	CloseEditor()
Python Example	`oDefinitionEditor.CloseEditor()`

Copy (Schematic Editor)

Copies items for pasting.

UI Access	**Edit** > **Copy**
Parameters	Name | Type | Description <NAME:Selections> | String | `"NAME:Selections"` <*Page*> | Integer | Page number (one-based), [Opt=1] Note: Page 1 always exists <*Selections*> | Array | List of ID strings to edit
Return Value	None

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting 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	Copy( )
Python Example	`oEditor.Copy (["NAME:Selections", "Selections:=", ["CompInst@R;2;37", "CompInst@C;1;1"]])`

CopyData [Schematic Editor]

Use: Copy graphic object or component properties for pasting.

Command: **Edit> Copy Data** (or Right-click on graphic object or component and select **Copy Data**)

Syntax: CopyData Array("NAME:Selections", "Selections:=", Array(<selections>))

Return Value: None.

Parameters: `<selections>`

Type: Comma-separated list of strings

Identifiers for each selected item to be copied, in the form CompInst@<id>, where id is the compname;ID;schematicID for each element.

CopySubdesign [Schematic Editor]

Copy components that represent subdesigns, as well as their represented designs, for pasting. The designs, when pasted, will be independent, with unique names based on the original design.

UI Access	(Right-click on hierarchical component) **Copy as New Design**
Parameters	

Name	Type	Description
<selections>	Comma-separated list of strings	Identifiers for each selected item to be copied, in the form CompInst@<id>, where id is the compname;ID;schematicID for each element.

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	CopySubdesign (["NAME:Selections", "Selections:=", [<selections>]])

Python Example	`oEditor.CopySubdesign ([`
	`"NAME:Selections", "Selections:=", [`
	`"CompInst@Simplorer2;23;17"]])`

CreatePage (Schematic Editor)

Create a page at the end of the existing pages

UI Access	Schematic>New Page
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><PageName></td><td>String</td><td>Name of the page that has been created</td></tr><tr><td><PageNum></td><td>Integer</td><td>Page Number</td></tr></table>
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting 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	CreatePage(string name, // Page name
	[out,retval] int num) // Page number

Python Example	`oEditor.CreatePage ("TestPage1")`

Cut (Schematic Editor)

Cut Page Selection

UI Access	Edit>Cut
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*Selections*></td><td>Array</td><td>`["NAME:Selections",` "Page:=", page number, _ // [Opt=1] Page number "Selections:=", [IDs to modify]]</td></tr></table>
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Cut(<*Selections*>)
Python Example	`oEditor.Cut (["NAME:Selections", "Selections:=", ["CompInst@R;10;93"]])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DeactivateOpen (Schematic Editor)

Disable items and replace with an open circuit

UI Access	**Edit > Deactivate (Open)**		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify]]
Return Value	None		

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	DeactivateOpen()
Python Example	oEditor.DeactivateOpen (["NAME:Selections", "Selections:=", ["CompInst@R;2;37"]])

DeactivateShort (Schematic Editor)

Disable items and replace with a closed circuit

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 > Deactivate (Short)**		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify]]
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	DeactivateShort ()
Python Example	oEditor.DeactivateShort(["NAME:Selections", "Selections:=", ["CompInst@R;2;37"]])

Delete (Schematic Editor)

Delete items.

UI Access	**Edit > Delete**		
Parameters	Name	Type	Description
	<Selections>	Array	["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.

			"Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify]]
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Delete()
Python Example	oEditor.Delete (["NAME:Selections", "Selections:=", ["CompInst@R;10;93"]])

DeletePage (Schematic Editor)

To reduce the number of pages in the schematic

UI Access	Schematic>Remove Page		
Parameters	Name	Type	Description
	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.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	DeletePage(<*pagenum*>, <*pagenum*>*)
Python Example	`num = oEditor.DeletePage([1, 4])`

ElectricRuleCheck (Schematic Editor)

Check Electric Rule

UI Access	**Schematic > Electric Rule Check**		
Parameters	Name	Type	Description
	<ElectricRules>	Array	Array of strings containing names of electric rules to perform. Valid names are: `"PinRule"`: Checks for components with unconnected pins `"OutputPinRule"`: Checks for nets with more than one output pin. `"OverlapCompRule"`: Checks for nets that overlap
	<*CheckCircuit*>	Boolean	`True` if this should check all subcircuits; `False` if this should just check this circuit
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ElectricRuleCheck(<*ElectricRules*>, <*CheckCircuit*>)
Python Example	`oEditor.ElectricRuleCheck (["PinRule", "OutputPinRule", "OverlapCompRule"], True)`

ExportImage (Schematic Editor)

To export a picture for a specified page of the current design to a file. The image size can also be specified. The filename extension determines the type of image exported.

UI Access	None.		
Parameters	Name	Type	Description
	<filename>	String	The name of the file, with format-specific extension. Extensions supported are: bmp, gif, jpg, jpeg, png, tif, tiff.
	<pagenum>	Integer	The page number of the current schematic. Page numbers start at 1. If the number is out of range, the current schematic page will be printed.
	<dx>	Integer	The width of the image. If `dx` is less than 160, 160 will be used for the width.
	<dy>	Integer	The height of the image. If `dy` is less than 160, 160 will be used for the height.
Return Value	None		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ExportImage (<filename>, <pagenum>, <dx>, <dy>)
Python Example	`oEditor.ExportImage ("c:\mysch.png", 1, 800, 400)`

FindElements (Schematic Editor)

Select elements based on properties

UI Access	Edit>Find Elements

Parameters

Name	Type	Description
<propname>	String	The name of the property, or "*" to search through all properties
<propvalue>	String	The value of the property, or a substring
<criterion>	Integer	0 = Contains 1 = Does not contain 2 = Exact match
<types>	Integer	2 - components only 4 - graphic objects only 8 - wires only 16 - ports only Add the numbers to look through more types - 30 is a common number, search-

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			ing through everything
	<match>	Boolean	True = match all of the properties specified False = match any of the properties specified
	<case>	Boolean	True = case-sensitive False = not case-sensitive
	<sub>	Boolean	True = search subcircuits as well as current schematic False = only search current schematic
	<only>	Boolean	True = search within current selections False = search schematic(s)
Return Value	Array of strings, the names of the elements selected		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	FindElements(["NAME:SearchProps", "Prop:=", [<propname>, <propvalue>, <criterion>)[, [<propname>, <propvalue>, <criterion>]]*], ["NAME:Parameters", "Filter:=", <types>, "MatchAll:=", <match>, "MatchCase:=", <case>, "SearchSubCkt:=", <sub>, "SearchSelectionOnly:=", <only>])

Python Example	`oEditor.FindElements(`
	`[`
	`"NAME:SearchProps",`
	`"Prop:=", ["CompName","",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:Parameters",
                "Filter:="              , 30,
                "MatchAll:="            , False,
                "MatchCase:="           , False,
                "SearchSubCkt:="        , True,
                "SearchSelectionOnly:=" , False
        ])
```

FitToBorder [Schematic Editor]

Use: To reset the schematic view to just include the page border, if any.

UI Access	View>Fit Border
Parameters	Name | Type | Description None | |
Return Value	None

Python Syntax	FitToBorder()
Python Example	`oEditor.FitToBorder()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

FlipHorizontal (Schematic Editor)

Flip items about the horizontal (x) axis

UI Access	NA		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, _ // [Opt=1] Page number "Selections:=",[IDs to modify]]
	<FlipParameters>	Array	["NAME: FlipParameters", "Disconnect:=", bool // [Opt=0] Should wires disconnect "Rubberband:=", bool]// [Opt=1] Should wires staircase
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	FlipHorizontal(<*Selections*>,<*FlipParameters*>)
Python Example	`oEditor.FlipHorizontal(`

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:=", ["CompInst@R;2;4", "CompInst@C;1;1"]], ["NAME:FlipParameters", "Disconnect:=", true, "Rubberband:=", false])

FlipVertical (Schematic Editor)

Flip items about the vertical (y) axis

UI Access	NA		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify)]]
	<FlipParameters>	Array	["NAME:FlipParameters", "Disconnect:=", bool // [Opt=0] Should wires disconnect "Rubberband:=", bool] // [Opt=1] Should wires staircase
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting 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	FlipVertical(<*Selections*>, <*FlipParameters*>)

Python Example	`oEditor.FlipVertical(`
	`  ["NAME:Selections",`
	`  "Selections:=", ["CompInst@R;2;4",`
	`  "CompInst@C;1;1"]],`
	`  ["NAME:FlipParameters",`
	`  "Disconnect:=", true, "Rubberband:=", false])`

GridSetup (Schematic Editor)

Changes the settings on the schematic grid

UI Access	**Schematic** > **Grid Setup**		
Parameters	Name	Type	Description

	<Options>	Array	`["NAME:Options",`
			`  "MajorGrid:=", string, // Major grid size with units`
			`  "Divisions:=", int, // Number of minor grid divisions`
			`  "MajorColor:=", int, // RGB Color of major grid lines`
			`  "MinorColor:=", int, // RGB Color of minor grid lines`
			`  "ShowGrid:=", bool, // Should the grid be shown?`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"SnapToGrid:=", bool, _// Should objects snap to grid? "BackgroundColor:=", int, _// RGB Color of the background "SaveAsDefault:=", bool]
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	GridSetup()

Python Example	`oEditor.GridSetup(`
	`  ["NAME:Options",`
	`  "MajorGrid:=", "10mm",`
	`  "Divisions:=", 10,`
	`  "MajorColor:=", 0x00ff0000, # Red`
	`  "MinorColor:=", 0x0000ff00, # Green`
	`  "ShowGrid:=", true,`
	`  "SnapToGrid:=", true,`
	`  "BackgroundColor:=", 0x00ffffff, \ # White`
	`  "SaveAsDefault:=", false`
	`])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Move (Schematic Editor)

Move items

<table>
<tr><td>UI Access</td><td>NA</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><elements></td><td>Array</td><td>["NAME:elements",
list of string IDs to move]</td></tr>
<tr><td><MoveParameters></td><td>Array</td><td>["NAME:MoveParameters",
"xdelta:=", double, # X distance to move
"ydelta:=", double _ # Y distance to move
"Disconnect:=", bool
[Opt=0]] #Should wires disconnect
"Rubberband:=", bool] // [Opt=1] Should wires staircase</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Move(<*elements*>, <*MoveParameters*>)
Python Example	`oEditor.Move (["NAME:elements", "circle_0", "rect_2"],`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	[0.0165613577023499, -0.001])

NameNets (Schematic Editor)

Change names of nets on the schematic

UI Access	Schematic>Auto-Name Wires
Parameters	Name | Type | Description None | |
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	NameNets(bool display, // Show Net names or not ["NAME:NetNames", _ "string:=", _ #Old name of the net (e.g. net_1) string, _ #New name of the net (e.g. my_net) ...]) # Repeat for additional nets to change
Python Example	`oEditor.NameNets (True, ["NAME:NetNames",` `"Wire@net_2;41:1:=", "A[0]"])`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PageBorders (Schematic Editor)

> **Note:**
>
> This command has been replaced with the SetPageData command. Legacy scripts using this command will play back for the first page of a schematic only.

Sets up visible page borders on the schematic

```
PageBorders(
Array("NAME:Options", _
"PageSize:=", _
Array("x:=", string, _ // Width of page border with units
"y:=", string), _ // Height of page border with units
"PageMargins:=", _
Array("x:=", string, _ // Margin Width with units
"y:=", string), _ // Margin Height with units
"ZonesHoriz:=", int, _ // Number of Horizontal zones
"ZonesVert:=", int)) // Number of Vertical zones
```

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Pan (Schematic Editor)

Pan Display

UI Access	View>Pan
Parameters	Name | Type | Description None | |
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Pan([double, _ // Move the visible area by X meters double]) // Move the visible area by Y meters
Python Example	`oDefinitionEditor.Pan` `([-0.00922653869663931, 0.00686839904222147])`

Paste (Schematic Editor)

Paste copied items

UI Access	Edit>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>None</td><td></td><td></td></tr></table>
Return Value	None

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	Paste(VARIANT attrs)# specified attributes

Python Example	`oEditor.Paste (["NAME:Attributes",`
	`"Page:=", 1, "X:=", -0.059131200000001, "Y:=",_`
	`0.030784799999999, "Angle:=", 0, "Flip:=",false])`

PasteData [Schematic Editor]

Use: Paste graphic object or component properties to another object or component.

Command: **Edit> Paste Data** (or Right-click on graphic object or component and select **Paste Data**)

Syntax: PasteData Array("NAME:Selections", "Selections:=", Array(<selections>), "SelectList:=", Array(<select_prop_display>, <select_parameters>), "Exclude:=", Array(<ExcludeList>))

Return Value: None.

Parameters: `Selections`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: Comma separated component instance list where user wants to paste data

Identifiers for each selected item to be copied, in the form CompInst@<id>, where id is the compname;ID;schematicID for each element.

`SelectList`

Type: Comma separated Boolean list where:

<select_prop_display> true to paste property displays

<select_parameters> true to paste parameters

`<ExcludeList>`

Type: Comma separated list of property names to exclude from being pasted.

PasteDesign (Schematic Editor)

Paste a design that has already been copied to the clipboard into schematic as a subdesign.

Syntax:

```
PasteDesign(
pasteOption , // see below for values
Array("NAME:Attributes", _
"Page:=", int, _ // [Opt=1] Page number (one-based)
// Note: Page 1 always exists
"X:=", double, _ // X position of the object
"Y:=", double, _ // Y position of the object
"Angle:=", double, _ // Rotation angle (radians)
"Flip:=", bool) // True if mirrored
)
```


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: Component is created and the returned value has the format CompInst@<CompName>;<compInstID>;<optional schematicID>

For instance a return value CompInst@CAP_;4;35 indicates that the component name is CAP_, its ID is 5, and its Schematic ID is 35. These details can be found in the Properties window when you select the created component. The component name and ID can be found in the Gen-eral tab and the Schematic ID can be found in the Symbols tab.

Parameters: pasteOption should be one of the following:

0 to link to the existing design that was copied

1 to create a new copy of the design that was copied but keep the original layers of the design being copied

2 to create a new copy of the design and merge the layers of the design being copied into the design receiving the copy

PushExcitations

Allows access to computed excitations for transient and linear frequency solutions. The script command can be accessed from three locations.

<table>
<tr><td>UI Access</td><td>

Layout Editor
Schematic Editor
Select a Nexxim solution in a 3D Layout design, right click, and choose Push Excitations.

</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td><Reference Designation></td><td>String</td><td>"<refdes>"

This argument is empty when excitations are pushed from a Nexxim solution.</td></tr>
<tr><td><PushExcitations Parameters></td><td>Array</td><td>This parameter changes depending on whether the excitations comes from a transient or linear frequency solution. The keyword</td></tr>
</table>
</td></tr>
</table>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			"transient:=" indicates to AEDT which solution generated the excitations. If "transient:=" is present, "CalcThevenin =" and its value are ignored. For a linear frequency solution, use this array: ["NAME:options", "CalcThevenin =" , <true|false>, "Sol:=", "<solution name>"] If CalcThevenin is true, Thevenin's equivalent is calculated. Parameters for the linear frequency solution do not include a Freq argument, so all frequencies from the solution are used. For a transient solution, use: ["NAME:options", "transient:=", ["start:=", <start time>, "stop:=", <stop time>, "maxHarmonics:=", <max harmonics>, "winType:=", <window>, ["widthPct:=", <width percentage>,] ["kaiser:=", <Kaiser value>,] ["correctCoherentGain:=", true]], "Sol:=", "<solution name>"] winType can have the following values: • Rectangular

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

			• Bartlett • Blackman • Hamming • Hanning • Kaiser • Welch • Weber • Lanzcos
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Rotate (Schematic Editor)

Rotate items

UI Access	Draw>Rotate		
Parameters	Name	Type	Description
	<Selections>	Array	["NAME:Selections", "Page:=", page number, // [Opt=1] Page number "Selections:=", [IDs to modify)]
	<RotateParameters>	Array	["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.

			"Disconnect:=", bool _ // [Opt=0] Should wires dis-connect "Rubberband:=", bool _ // [Opt=1] Should wires stair-case "Angle:=", double] // [Opt=/2] Angle to rotate Note: Rotation occurs around center of ids
Return Value	None		

Note:

When you record a script, the schematic editor's Rotate command shows the rotation angle for some older scripts in radians, while newer scripts show rotation angle in degrees.

```
oEditor.Rotate ["NAME:Selections",
  "Selections:=", ["CompInst@RES_;1;1:1"]],
  ["NAME:RotateParameters",
    "Angle:=", 1.5707963267949, "Disconnect:=", false, "Rubberband:=", false]
```

If "Degrees=xxx" is not shown, the rotation must be scaled.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting 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	Rotate()
Python Example	`oEditor.Rotate (["NAME:Selections", "Selections:=", ["CompInst@R;2;4", "CompInst@C;1;1"]], ["NAME:RotateParameters","Degrees:=", 90, "Disconnect:=", _ false, "Rubberband:=", false])`

SelectAll (Schematic Editor)

Select all elements on the given page.

UI Access	Edit>Select All
Parameters	Name | Type | Description None | |
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	SelectAll(int pageNum) # Page number The first page number is 1. If an invalid page number is passed in (i.e. 0, -1, etc), `SelectAll` will use the active page on the currently active view. If there is no active view, the first page, page 1, is used.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Example	`oEditor.SelectAll(1)`

SelectPage (Schematic Editor)

Select a page in the UI

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	SelectPage(int page) # Page number
Python Example	`oEditor.SelectPage(1)`

SendToBack

Send the selected object to the back of the view

UI Access	Draw>Send To Back
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.

	<Object>	String	Object to send to the back.
Return Value	None		

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs

Python Syntax	SendToBack (["NAME:Selections", "Selections:=", [<Object>, <Object>, ...]])
Python Example	`oDefinitionEditor.SendToBack(["NAME:Selections",` `"Selections:=", ["SchObj@10"]])`

SetPageData [Schematic Editor]

> **Note:**
>
> This command replaces the **PageBorders** command.

To add or remove page border, title block and page property data from the specified page. Title blocks will not be shown if the page border type is "None". Any number of title blocks may be specified, as well as any number of page properties. Some of the page properties are special and their values will be updated automatically as appropriate, such as Project, ProjectPath, Design, Number and Date.

UI Access	**Schematic** > P**age Borders and Title Blocks**...

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
<BorderArray>		
<*bordertype*>	String	One of: "None", "Outline", "ANSI", "ISO", "DIN"
<*sizeX*>	Double (with units)	Size of X axis
<*sizeY*>	Double (with units)	Size of Y axis
<*marginX*>	Double (with units)	Margin of X axis
<*marginY*>	Double (with units)	Margin of Y axis
<*zonesH*>	Integer	
<*zonesV*>	Integer	
<TitleBlockArray>		
<*symbolName*>	String	The name of the title block symbol.
<*libName*>	String	The name of the library (with path from library directory root) that contains the title block symbol
<*Hsnap*>	String	One of: "Left", "Right", "Center", "None"
<*Vsnap*>	String	One of "Top", "Bottom", "Center", "None"
<*rotation*>	String	One of "0", "90", "270"
<PagePropsArray>		
<*propName*>	String	The name of a page property followed by ":="
<*propVal*>	String	The value of a page property

Return Value

None

Python Syntax

SetPageData (<*pagenum*>,<BorderArray>,<TitleBlockArray>, <PagePropsArray>)

Python Example

```
oEditor.SetPageData (1,
  ["NAME:Border",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"BorderType:=", "ANSI", "PageSize:=", ["x:=", "11in", "y:=", "8.5in"], "PageMargins:=", ["x:=", "0.38in", "y:=", "0.25in"], "ZonesHoriz:=", 2, "ZonesVert:=", 2], ["NAME:TitleBlocks", "TitleBlock:=", ["TitleBlk1","TitleBlocks", "Right", "Bottom", "0", -1]], ["NAME:PageProps", "Title:=", "Design Title", "Author:=", "I. M. User", "Number:=", "1", "Project:=", "Project8", "ProjectPath:=", "e:/projects/", "Design:=", "TwinBuilder1", "Date:=", "9:53:11 PM Jan 27, 2020"])

ShowVariableBlock (Schematic Editor)

Controls display of Variable Text Block in Schematic Editor.

UI Access	From the Schematic Ribbon, select **Settings** > **Design Variables**
Parameters	Name | Type | Description *<toggle>* | Boolean | `True` displays the Text Block; `False` hides the Text Block.
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	ShowVariableBlock ()
Python Example	`oEditor.ShowVariableBlock("True")`

SortComponents (Schematic Editor)

Sorts all components, or a block of components, using the specified method

UI Access	Schematic>Sort Components

Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><Type></td><td>Array</td><td>[type, // string specifying sort type
	method] // string specifying sort method type
	// string argument specifying sort type: "All Components" or "Blocks"
	// An empty string corresponds to "All Components" method
	// string argument specifying sort method: "By Name", "Left to Right", or "Signal Flow"</td></tr></table>

Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	SortComponents ()
Python Example	`oEditor.SortComponents()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Wire (Schematic Editor)

Wire specified pairs of pins in two selected component instances.

UI Access	Select two component instances in the Schematic Editor, then right click on one and choose **Wire**.
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><*firstComponent*></td><td>String</td><td>Name of the first component to be wired</td></tr><tr><td><*firstListOfPins*></td><td>String</td><td>List of pins separated by spaces</td></tr><tr><td><*secondComponent*></td><td>String</td><td>Name of second component the first component will be wired to</td></tr><tr><td><*secondListOfPins*></td><td>String</td><td>List of pins on the second component, separated by spaces</td></tr></table>
Return Value	None

Python Syntax	Wire(<*firstComponent*>,<*firstListOfPins*>,<*secondComponent*>,<*secondListOfPins* >)
Python Example	`oEditor.Wire("CompInst@etch_sline_s20_z70_1;10;76", "A1 A2", "CompInst@CX0805MRNPO9BB220;1;1", "B1 B2")`

ZoomArea (Schematic Editor)

Zoom Area

UI Access	View>Zoom Area
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><Size></td><td>Array</td><td>[double, double] //Base X value of the area to zoom into double),</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.

			// Base Y value of the area to zoom into
			[double,double]
			// X size of the area to zoom
			// Y Size of the area to zoom
			NOTE: All values in meters

Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ZoomArea()
Python Example	`oEditor.ZoomArea ([0.00307569800375911, 0.021101611633739], [-0.050433530091516, -0.0220260723256825])`

ZoomIn (Schematic Editor)

Zoom into the schematic

UI Access	View>Zoom In		
Parameters	Name	Type	Description
	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.

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ZoomIn()
Python Example	`oEditor.ZoomIn()`

ZoomOut (Schematic Editor)

Zoom out from the schematic

UI Access	View>Zoom Out
Parameters	Name | Type | Description None | |
Return Value	None

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ZoomOut()

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.ZoomOut()`

ZoomPrevious (Schematic Editor)

Restore the zoom to the previous settings

UI Access	NA
Parameters	Name | Type | Description None
Return Value	None

> **Note:**
>
> For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ZoomPrevious()
Python Example	`oEditor.ZoomPrevious()`

ZoomToFit (Schematic Editor)

Set the Zoom to Fit

UI Access	**View > Fit All**
Parameters	Name | Type | Description

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	None		
Return Value	None		

Note:

For more information about the component name, schematic IDs, and formats, see the section Editor Scripting IDs.

Python Syntax	ZoomToFit()
Python Example	`oEditor = oDesign.SetActiveEditor("SchematicEditor")` `oEditor.ZoomToFit()`

Property Method List

This section presents the property script methods that are available.

ChangeProperty (Schematic Editor)

GetEvaluatedPropertyValue (Schematic Editor)

GetProperties (Schematic Editor)

GetPropertyAttribute [Schematic Editor]

GetPropertyValue (Schematic Editor)

SetPropertyValue (Schematic Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeProperty (Schematic Editor)

Changes a property in the schematic editor using the `ChangeProperty` method.

> **Note:** For a complete list of supported parameters and tab types, refer to ChangeProperty (Schematic Editor and Layout Editor).

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><NAME:AllTabs></td><td>Array</td><td>Top-level wrapper for all tab-specific property changes.</td></tr><tr><td><NAME:ComponentTab></td><td>Array</td><td>Specifies the tab being modified.</td></tr><tr><td><NAME:PropServers></td><td>Array</td><td>Identifies the target object(s) whose properties are being modified (e.g., <code>"Wire@net_1;14:1"</code>)</td></tr><tr><td><NAME:ChangedProps></td><td>Array</td><td>Contains the list of properties to be changed for the specified PropServer.</td></tr><tr><td><NAME:NetName></td><td>String</td><td>Name of the net being modified.</td></tr><tr><td><ExtraText></td><td>String</td><td>Additional label or annotation for the net.</td></tr><tr><td><SplitWires></td><td>Boolean</td><td>True – split wires for the net. False – do not split wires for the net.</td></tr></table>
Return Value	None

Python Syntax	ChangeProperty()

Python Example	`oEditor.ChangeProperty(["NAME:AllTabs",`
	`["NAME:ComponentTab", ["NAME:PropServers",`
	`"Wire@net_1;14:1"], ["NAME:ChangedProps",`
	`["NAME:NetName", "ExtraText:=",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`"net_10", "Name:=", "net_10", "SplitWires:=", false]]])`

GetEvaluatedPropertyValue (Schematic Editor)

Get value.

UI Access	**Not available**
Parameters	Name / Type / Description *<tabDescription>* / String / Name of property tab where property is found *<componentID>* / String / ID of component instance where property is found, in the format: "CompInst@<name of component type>;<id of compInstance>" *<propName >* / String / Name of the variable property
Return Value	None

> **Note:**
>
> This function is only available with the schematic editor.
>
> Calling this function on non-numeric properties (e.g. Text properties) returns 0.

Python Syntax	GetEvaluatedPropertyValue(*<tabDescription*, *<componentID>*, *<propName>*)
Python Example	`evalValue = oEditor.GetEvaluatedPropertyValue("PassedParameterTab", "CompInst@CAP_;3", "C")`

GetProperties (Schematic Editor)

To get the names of properties for a specified tab and element.

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>NA</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><tab></td><td>String</td><td>One of six choices, "PassedParameterTab", "ComponentTab", "Component", "BaseElementTab", "Quantities", "Signals"</td></tr>
<tr><td><objectId></td><td>String</td><td>Instance ID of object with the property</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>Array of strings, the property names</td></tr>
</table>

Python Syntax	GetProperties(<tab>, <objectId>)

Python Example	`comparray = oEditor.GetAllComponents()`
	`for i in comparray:`
	`  instval = oEditor.GetPropertyValue`
	`  ("PassedParameterTab",i,"InstanceName")`
	`  proparray = oEditor.GetProperties ("Quantities",i)`
	`  for j in proparray:`
	`  direction = oEditor.GetPropertyAttribute`
	`  ("Quantities",i, j, "Direction")`
	
	`  AddInfoMessage(str(j) + " = (direction) " +`
	`  str(direction))`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropertyAttribute [Schematic Editor]

To find the value of an attribute of a specified property.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><tab></td><td>String</td><td>One of six choices, "PassedParameterTab", "ComponentTab", "Component", "BaseElementTab", "Quantities", "Signals"</td></tr><tr><td><objectId></td><td>String</td><td>Instance ID of object with the property</td></tr><tr><td><propName></td><td>String</td><td>The name of the property</td></tr><tr><td><attribute></td><td>String</td><td>Currently, only the "Direction" attribute is supported. Return values will be one of "In", "Out" or "InOut"</td></tr></table>
Return Value	A string, the attribute's value.

Python Syntax	GetPropertyAttribute(<tab>, <objectId>, <propName>, <attribute>)

Python Example	`comparray = oEditor.GetAllComponents()`
	`for i in comparray:`
	`instval = oEditor.GetPropertyValue`
	`("PassedParameterTab",i,"InstanceName")`
	`proparray = oEditor.GetProperties ("Quantities",i)`
	`for j in proparray:`
	`direction = oEditor.GetPropertyAttribute`
	`("Quantities",i, j, "Direction")`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`AddInfoMessage(str(j)+ " = (direction) " +` `str(direction))`

GetPropertyValue (Schematic Editor)

Get a property

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><tab></td><td>String</td><td>Tab with the property</td></tr><tr><td><item></td><td>String</td><td>Name of the object</td></tr><tr><td><propname></td><td>String</td><td>Name of the property</td></tr><tr><td><value></td><td>String</td><td>Value of the property</td></tr></table>
Return Value	None

Python Syntax	GetPropertyValue(<tab>,<item>,<propname>,<value>)
Python Example	None

SetPropertyValue (Schematic Editor)

Set a property.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><tab></td><td>String</td><td>Tab with the property</td></tr><tr><td><item></td><td>String</td><td>Name of the 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.

	<propname>	String	Name of the property
	<value>	String	The new value of the property
Return Value	None		

Python Syntax	SetPropertyValue(<params>)
Python Example	None

Information Method List

This section presents the information methods that are available.

GetCompInstanceFromRefDes (Schematic Editor)

GetComponentInfo (Schematic Editor)

GetComponentPins (Schematic Editor)

GetComponentPinInfo (Schematic Editor)

GetComponentPinLocation [Schematic Editor]

GetEditorName (Schematic Editor)

GetNetConnections (Schematic Editor)

GetNumPages [Schematic Editor]

GetPortInfo (Schematic Editor)

GetSelections (Schematic Editor)

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetSignals (Schematic Editor)

GetAllPorts

Get all ports in a design.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Ports in the design

Python Syntax	GetAllPorts()
Python Example	`oEditor.GetAllPorts()`

GetCompInstanceFromRefDes (Schematic Editor)

Informational

UI Access	NA
Parameters	Name | Type | Description Component | String | Name of the Component
Return Value	Returns IDispatch for CompInstance

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetCompInstanceFromRefDes(string) # string is refDes for the component
Python Example	`GetCompInstanceFromRefDes("R1")`

GetComponentInfo (Schematic Editor)

Informational

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><compID></td><td>String</td><td>The Schematic component's ID, obtained from GetSelections or through FindElements</td></tr></table>
Return Value	Array of Strings, as follows: "Page=*number*" "ComponentName=*string*" "GateCount=*number*" "LocationX=*number*" "LocationY=*number*" "BBoxLLx=*number*" "BBoxLLy=*number*" "BBoxURx=*number*"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"BBoxURy=*number*" "Angle=*number*" (in degrees) "Flip=true *or* false"

Python Syntax	GetComponentInfo(<compID>)

Python Example	`For i in selectionArray:`
	`  compArray = oEditor.GetComponentInfo(i)`
	`  For Each k in compArray:`
	`  AddInfoMessage(str(k) +`
	`  "Got with GetComponentInfo")`

GetComponentPins (Schematic Editor)

Informational

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><compID></td><td>String</td><td>The Schematic component’s ID, obtained from GetSelections or through FindElements.</td></tr></table>
Return Value	Array of strings, which are the names of all the component's pins

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetComponentPins(<compID>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for i in selectionArray`
	`  pinArray = oEditor.GetComponentPins(i)`
	`  for j in pinArray`
	`  AddInfoMessage("Pin: " + str(j))`

GetComponentPinInfo (Schematic Editor)

Informational - gets details about a pin on a schematic component.

UI Access	NA		
Parameters	Name	Type	Description
	<compID>	String	The Schematic component's ID
	<pinName>	String	The pin name
Return Value	String values, as follows: "X=number" "Y=number" "Angle=number" (in degrees) "Flip=true or false" "WireId=netID or empty string" (netID is the identifier of the schematic net, and can be used in other informational methods)		

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetComponentPinInfo(<compID>, <pinName>)
Python Example	`info = oEditor.GetComponentPinInfo` `("1","N1")`

GetComponentPinLocation [Schematic Editor]

Informational - gets the X or Y location of a pin on a schematic component. Invoke once with TRUE for "XorY" to get the X value of the pin's location in the schematic, and once with FALSE to get the Y value.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><compID></td><td>String</td><td>The Schematic component’s ID</td></tr><tr><td><pinName></td><td>String</td><td>The pin name</td></tr><tr><td><XorY></td><td>Boolean</td><td>TRUE gets the X value; FALSE gets the Y value.</td></tr></table>
Return Value	String values, as follows: "X=*number*" "Y=*number*"

Python Syntax	GetComponentPinLocation <compID>, <pinname>, <XorY>
Python Example	`varx = oEditor.GetComponentPinLocation`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`(compid, "N1", TRUE)`

GetEditorName (Schematic Editor)

Get the name of the schematic editor.

UI Access	N/A
Parameters	None
Return Value	String containing the name of the schematic editor. “SchematicEditor”

Python Syntax	GetEditorName()
Python Example	`oEditor.GetEditorName()`

GetNetConnections (Schematic Editor)

Informational.

UI Access	Not available
Parameters	Name | Type | Description *<netName>* | String | The name of the net. Specifying "0" as the netName will return all objects connected to ground including all ground ports.
Return Value	Array of strings containing the connections for the net identified by the netName argument. Strings in the array

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	are in one of four formats: "IPort portID x y pinName" "GPort portID x y pinName" "PagePort portID x y pinName" "Component componentName compID x y pinName" where *IPort, GPort, PagePort, and Component* are key words that indicate what the rest of the string specifies: *IPort* indicates that *portID* specifies the name of the interface port, *GPort* indicates that *portID* specifies the name of the global port, *PagePort* indicates that *portID* specifies the name of the page port, *Component* indicates that *compID* specifies the component instance identifier, *x* and *y* are the connection point, *pinName* is the name of the connected pin, *componentName* is the name of the component.

Python Syntax	GetNetConnections(<netName>)
Python Example	`netArray = oEditor.GetNetConnections("net_1")`

GetNumPages [Schematic Editor]

To determine how many pages there are in the schematic.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Integer, the number of pages in the schematic.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetNumPages()
Python Example	`num = oEditor. GetNumPages()`

GetPortInfo (Schematic Editor)

Informational – get details about a port on a schematic component.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><portID></td><td>String</td><td>The Schematic port's ID, obtained from GetSelections or through FindElements.</td></tr></table>
Return Value	Array of strings, as follows: "Name=string" // Name of the port. "X=val" // Connection point X location. "Y=val" // Connection point Y location. "Angle=val" // Component angle. "Flip=true or false" // Whether the component is flipped or not. "WireId=string" // The identifier of the schematic net, can be used in other informational methods.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetPortInfo(<portID>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for i in selectionArray:`
	`  info = oEditor.GetPortInfo(i)`
	`for j in info:`
	`  AddInfoMessage(str("GetPortInfo: " + j))`

GetSelections (Schematic 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()`

GetSignals (Schematic Editor)

Informational

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UI Access	NA
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><wirename></td><td>String</td><td>The wire's name, as seen in the UI, obtained in script through parsing return data from `GetWireInfo`.</td></tr></table>
Return Value	Array of strings which contains the signal names in the *wirename* argument. The *wirename* "Net1" would return an array with a single string "Net1". The *wirename* "WW[0-3],A,B" would return an array with six strings "WW[0]", "WW[1]", "WW[2]", "WW[3]", "A", :B".

Python Syntax	GetSignals(<wirename>)

Python Example	```
	selectionArray = oEditor.GetSelections()
	for i in selectionArray:
	
	  netArray3 = oEditor.GetWireInfo(i)
	  for n in netArray3:
	
	  wname = n[0:9]
	
	  if (wname == "WireName:"):
	  nn = len(n)
	  wn = n[nn-10:]
	  AddInfoMessage("Wirename, parsed from ID :" + str(wn))
	  sigs = oEditor.GetSignals(wn)
	``

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	for ll in sigs: AddInfoMessage("Element in array from GetSignals :" + str(ll))

GetWireConnections (Schematic Editor)

Informational.

UI Access	NA
Parameters	Name | Type | Description None | |
Return Value	Array of strings containing the connections for the wire identified by the wireID argument. The strings in the array are in one of four formats: "InterfacePort=portname portID x,y" "GlobalPort=portname portID x,y" "PagePort=portname portID x,y" "ComponentPin=compId pinname x,y" where x,y is the connection point.

Python Syntax	GetWireConnections(<wireID>)
Python Example	selectionArray = oEditor.GetSelections() for i 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.

```
        AddInfoMessage("Selected element" + i)
        netArray = oEditor.GetWireConnections(i)
        for m in netArray:
           AddInfoMessage("GetWireConnections" + str(m))
```

GetWireInfo (Schematic Editor)

Informational.

UI Access	NA		
Parameters	Name	Type	Description
	<wire id>	String	The Schematic wire's ID, obtained from `GetSelections` or through `FindElements`.

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 wire info in the wireID argument. The strings in the array are: "Page=number" "WireName=string" "SegmentCount=number" "IPortCount=number" "GPortCount=number" "PagePortCount=number" "PinCount=number" "BBoxLLx=val" "BBoxLLy=val" "BBoxURx=val" "BBoxURy=val"

Python Syntax	GetWireInfo(<wireID>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for i in selectionArray:`
	`  AddInfoMessage("Selected element" + str(i))`
	`  netArray2 = oEditor.GetWireInfo(i)`
	`  for n in netArray2:`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`AddInfoMessage("GetWireInfo" + str(n))`

GetWireSegments (Schematic Editor)

Informational.

UI Access	NA
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><wireID></td><td>String</td><td>The Schematic wire's ID, obtained from `GetSelections` or through `FindElements`</td></tr></table>
Return Value	Array of strings containing the wire segments in the wireID argument. The strings in the array are: "Segment=val,val val,val number" the "val" quantities are segment endpoints x1,y1 x2,y2 and the "number" is a schematic segment ID

Python Syntax	GetWireSegments (<wireID>)

Python Example	`selectionArray = oEditor.GetSelections()`
	`for i in selectionArray:`
	`  AddInfoMessage("Selected element" + str(i))`
	`  netArray2 = oEditor.GetWireSegments(i)`
	`  for n in netArray2:`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	`AddInfoMessage("GetWireSegments" + str(n))`

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.

17 - Nexxim Scripting

Nexxim scripting commands should be executed by the **oDesign** object.

For example,

oDesign.GetModule("DataBlock")

oModule.CommandName <args>

The topics for this section include:

Nexxim Netlist Scripting

Nexxim Data Block Commands

Nexxim Simulation Setup Commands

Nexxim Linear Network Analysis

Interface Ports And Sources Commands

Nexxim Component Manager Commands

Nexxim Netlist Scripting

The following sections present the Nexxim design level scripting commands that are available.

Analyze

CopyEyeItemAsCommand

DeleteDesignInstance

EditImportData

EditNotes

ExportForSpice

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportForHSpice

ExportNetlist

GetModule

GetName

GetEditor

GetResultsDirectory

ImportData

ImportDataFilePath

InsertDesign

PasteItemCommand

Redo

RenameDesignInstance

RenameImportData

SetActiveEditor

Undo

UseCircuitSParameterDefinition

Analyze [Nexxim]

Simulates the given setup

Syntax: Analyze(STRING setup1) // setup name

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	Analyze(<setupName>)
Python Example	oDesign.Analyze("QuickEyeAnalysis")

CopyEyeItemAsCommand

Use: Make a QuickEye copy of a VerifEye analysis, or a VerifEye copy of a QuickEye analysis.

Command: Select an analysis in the Project tree, then right-click and select **Copy As QuickEye** or **Copy As VerifEye**.

Syntax: CopyEyeItemAsCommand <itemPathList>

Return Value: Copy of the analysis.

Parameters: <itemPathList>

Type: Array of strings // Type of analysis from which to copy.

Python Syntax	CopyEyeItemAsCommand <itemPathList>
Python Example	oDesign.CopyEyeItemAsCommand(["eye_diagram_sch|Nexxim1|Analysis|VerifEyeAnalysis"]) oDesign.CopyEyeItemAsCommand(["eye_diagram_sch|Nexxim1|Analysis|QuickEyeAnalysis"])

DeleteDesignInstance

Use: Delete the design instance

Command: None

Syntax: DeleteDesignInstance <instance_name>

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <instance_name>

Type: string

Python Syntax	DeleteDesignInstance(<instanceName>)
Python Example	`oProject.DeleteDesignInstance("Circuit1")`

EditImportData

Use: Edit an imported solution

Command: None

Syntax: EditImportData <oldname> <newlink> <newpath> <newname> <data> <file>

Return Value: None

Parameters: <oldname>

Type: string

<newlink>

Type: int

<newpath>

Type: string

<newname>

Type: string

<data>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: array

<file>

Type: string

EditNotes

Updates the notes for the design.

UI Access	**Circuit** > **Edit Notes**.
Parameters	Name | Type | Description <*DesignNotes*> | String | New design notes that are applied, replacing any current notes.
Return Value	None.

Python Syntax	EditNotes(<*DesignNotes*>)
Python Example	`oDesign.EditNotes("My design notes.")`

ExportForSpice [Nexxim]

Export matrix solution data to a file in the given spice format.

Command: None

Syntax: ExportForSpice <theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fittingError> <maximumOrder> <useCommonGround> <doPassivityCheck>

Return Value: None

Parameters: <theVariationKey>

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.

<sourceNames>

Type: array

<theType: 0=PSpice, 2=MaxwellSpice, 3=Spectre>

Type: long

<bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: long

<fullwaveSpiceFilename>

Type: string

<lumpedElementFilename>

Type: string

<Unused>

Type: string

<Unused>

Type: string

<partialFractionFilename>

Type: string

<fittingError [Default:0.5]>

Type: double

<maximumOrder [Default:200]>

Type: int

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<useCommonGround [Default:FALSE]>

Type: boolean

<doPassivityCheck [Default:FALSE]>

Type: boolean

Python Syntax	ExportForSpice(<theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fittingError> <maximumOrder> <useCommonGround> <doPassivityCheck>)
Python Example	oDesign.ExportForSpice("", ["LNA:LNA"] , 0, 0, "C:/Documents and Settings/Marcia.BMT/MyDocuments/Code Modifications/Test" & " Designs/s-params_fws_fws.lib", "C:/Documents and Settings/Marcia.BMT/MyDocuments/Code Modifications/Test" & " Designs/s-params_fws_lfws.lib", "", "", "", 0.52,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	210, 1, 1)

ExportForHSpice [Nexxim]

Export matrix solution data to an HSpice subcircuit.

Command: None

Syntax: ExportForHSpice <theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fitting Error> <maximumOrder> <useCommonGround> <doPassivityCheck>

Return Value: None

Parameters: <theVariationKey>

Type: string

<sourceNames>

Type: array

<theType: 1=HSPICE>

Type:long

<bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: long

<fullwaveSpiceFilename>

Type: string

<lumpedElementFilename>

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

<Unused>

Type: string

<Unused>

Type: string

<partialFractionFilename>

Type: string

<fittingError [Default:0.5]>

Type: double

<maximumOrder [Default: 200]>

Type: int

<useCommonGround [Default: FALSE]>

Type: boolean

<doPassivityCheck [Default: FALSE]>

Type: boolean

Python Syntax	ExportForHSpice(<theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fitting Error> <maximumOrder> <useCommonGround> <doPassivityCheck>)
Python Example	`oDesign.ExportForHSpice( "", ["Setup1:Sweep 1"], 1, 0, "C:/Projects/MyTRL_fws.sp", "", "", "", "", 0.5, 0, 200, 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.

ExportNetlist [Nexxim]

Exports a netlist solution.

UI Access	None
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><*solution*></td><td>String</td><td>The Circuit solution in the design. For Maxwell Circuit, the solution must be an empty string, "".</td></tr><tr><td><*fileName*></td><td>String</td><td>File and path to the netlist file. For Maxwell Circuit, the file name must be a SPH file.</td></tr></table>
Return Value	None.

Python Syntax	ExportNetlist(<setupName>,<fileName>)
Python Example	`oDesign.ExportNetlist("QuickEyeAnalysis", "SolutionData")`

GetModule [Nexxim]

Get the IDispatch for the specified module.

Command: None

Syntax: GetModule <"modulename"> <object_variable>

Return Value: Module IDispatch

Parameters: <"modulename">

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.

<object_variable>

Type: string

Python Syntax	GetModule(<moduleName>,<object_variable>)
Python Example	`oDesign.GetModule("modulename" object_variable)`

GetName [Nexxim]

Get the name of the active design.

Command: None

Syntax: GetName <namestring_variable>

Return Value: String

Parameters: <namestring_variable>

Type: string

Python Syntax	GetName(<namestring_variable>)
Python Example	`oDesign.GetName(namestring_variable)`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetEditor [Nexxim]

Get the named editor without activating it.

Command: None

Syntax: GetEditor <editorName object_variable>

Return Value: String

Parameters: <editorName>

Type: string

<object_variable>

Type: strin

GetResultsDirectory [Nexxim]

Gives access to the directory containing the specified Nexxim solution files.

Command: None

Syntax: GetResultsDirectory ([in] BSTR solname, [in] BSTR variation, [in] BOOL finalDir, [out, retval] BSTR* dirname)

Return Value: <dirname>

Type: String // Directory name. (The returned directory path does not contain a trailing slash.).

Parameters: <solname>

Type: string

<variation>

Type: string

<finalDir>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Type: boolean // True if the final directory is desired; False if the working/temporary directory is desired.

Python Syntax	dirname = GetResultsDirectory (< solname>, <variation>,<finalDir>,)
Python Example	`dirName = oDesign.GetResultsDirectory("Trans", "res='1000'", False)`

ImportData [Nexxim]

Imports a network solution

Command: None

Syntax: ImportData <data> <filename> <linktofile>

Return Value: None

Parameters: <data>

Type: string // data to be imported

<filename>

Type: string // name of the file

<linktofile>

Type: int // link to the file

Python Syntax	ImportData(<data>,<file>, <customization>)
Python Example	`oDesign.ImportData(` `  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:NPortData",
      "Description:="           , "",
      "ImageFile:="             , "",
      "SymbolPinConfiguration:=", 0,
      [
         "NAME:PortInfoBlk"
      ],
      [
         "NAME:PortOrderBlk"
      ],
      "filename:="      , "E:/R18Examples/Signal Integrity/models/Serial_Chan-
nel/connector_model.s4p",
      "numberofports:=" , 4,
      "sssfilename:="           , "",
      "sssmodel:="              , False,
      "PortNames:="             , ["Port1","Port2","Port3","Port4"],
      "domain:="                , "frequency",
      "datamode:="              , "Link",
      "devicename:="            , "",
      "SolutionName:="  , "connector_model",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "displayformat:=" , "MagnitudePhase",
        "datatype:="              , "SMatrix",
        [
           "NAME:DesignerCustomization",
           "DCOption:="           , 0,
           "InterpOption:="       , 0,
           "ExtrapOption:="       , 1,
           "Convolution:="        , 0,
           "Passivity:="          , 0,
           "Reciprocal:="         , False,
           "ModelOption:="        , "",
           "DataType:="           , 1
        ],
        [
           "NAME:NexximCustomization",
           "DCOption:="           , 3,
           "InterpOption:="       , 1,
           "ExtrapOption:="       , 3,
           "Convolution:="                , 0,
           "Passivity:="          , 0,
           "Reciprocal:="         , False,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "ModelOption:="                , "",
        "DataType:="           , 2
    ],
    [
        "NAME:HSpiceCustomization",
        "DCOption:="           , 1,
        "InterpOption:="       , 2,
        "ExtrapOption:="       , 3,
        "Convolution:="        , 0,
        "Passivity:="          , 0,
        "Reciprocal:="         , False,
        "ModelOption:="        , "",
        "DataType:="           , 3
    ],
    "NoiseModelOption:="       , "External"
], "", False)
```

ImportDataFilePath [Nexxim]

Returns the file path of the imported solution given an imported solution name. If the solution is not found it returns an empty string.

Command: None

Syntax: ImportDataFilePath <solution_name>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: File path of imported solution, empty string if solution not found in the design.

Type: string

Parameters: <solution_name>

Type: string

Python Syntax	ImportDataFilePath(<solution_name>)
Python Example	`solution_path_variable = oDesign.ImportDataFilePath("MySolution")`

InsertDesign [Nexxim]

Insert a design

Command: None

Syntax: InsertDesign <type>, <name>, <stationerypath>, <parentname>

Return Value: None

Parameters: <type>

Type: string

<name>

Type: string

<stationerypath>

Type: string

<parentname>

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.

Python Syntax	InsertDesign(<typeName> <designName> ,<pathName>,<parentName>)
Python Example	`oProject.InsertDesign("Circuit Design", "Circuit1", "", "")`

PasteItemCommand

Paste tree items, such as Altrasim Solution and Solve Setups pasted to the Analysis Tree, or Frequency Sweeps pasted to the Solve Setup Tree.

Command: Right-click on the Analysis item in the Project tree and select Paste.

Syntax: PasteItemCommand <ItemPath>

Return Value: None

Parameters: <ItemPath>

Type: string

Python Syntax	PasteItemCommand(<ItemPath>)
Python Example	`oDesign.PasteItemCommand("Project1|Nexxim1|Analysis")`

Redo [Nexxim]

Redo the last operation

Command: Edit>Redo

Syntax: Redo

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Return Value: None

Parameters: None

Python Syntax	Redo()
Python Example	`oDesign.Redo()`

RenameDesignInstance [Nexxim]

Rename the design instance

Command: None

Syntax: RenameDesignInstance <oldname> <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Python Syntax	RenameDesignInstance(<oldName>,<newName>)
Python Example	`oDesign.RenameDesignInstance("Circuit1", "TestCircuit1")`

RenameImportData [Nexxim]

Rename an imported solution

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: RenameImportData <oldname> <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Python Syntax	RenameImportData(<oldName>,<newName>)
Python Example	`oDesign.RenameImportData("Solution", "ConnectorData")`

SetActiveEditor [Nexxim]

Set the active editor to the named one. (Activates window or opens window if necessary).

Command: None

Syntax: SetActiveEditor <editorName> <object_variable>

Return Value: None

Parameters: <editorName>

Type: string

<object_variable>

Type: string

Example: oDesign.SetActiveEditor <editorName> <object_variable>

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	SetActiveEditor(<editorName> <object_variable>)
Python Example	`oDesign.SetActiveEditor("Schematic" object_variable)`

Undo [Nexxim]

Undo the last operation

Command: **Edit>Undo**

Syntax: Undo

Return Value: None

Parameters: None

Example: `oDesign.Undo`

Python Syntax	Undo()
Python Example	`oDesign.Undo()`

UseCircuitSParameterDefinition

Selects whether the circuit or high-frequency definition of S-parameters is used.

Command: None

Syntax: UseCircuitSparameterDefinition <boolean_operator>

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <boolean_operator>

Type: bool

Python Syntax	UseCircuitSparameterDefinition(True)
Python Example	`oDesign.UseCircuitSparameterDefinition(True)`

Nexxim Data Block Commands

The following sections present the commands that are available in the "DataBlock" Module and can be used with the Module command interface:

For example,

```
oDesign.GetModule("DataBlock")
oModule.CommandName <args>
```

Nexxim data clock commands are listed below.

AddDeviceNoiseDataBlock

AddLibRefDataBlock

AddNetlistDataBlock

AddPrintToAuditDataBlock

AddStateVariableDataBlock

AddSubstrateDataBlock

AddTemperatureDataBlock

EditTemperatureDataBlock

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditLibRefDataBlock

EditNetlistDataBlock

EditPrintToAuditDataBlock

EditStateVariableDataBlock

EditSubstrateDataBlock

EditDeviceNoiseDataBlock

GetTemperatureDataBlock

GetAllLibRefDataBlocks

GetAllNetlistDataBlocks

GetAllSubstrateDataBlocks

Remove

Rename

AddTemperatureDataBlock

Adds a Temperature data block.

UI Access	None		
Parameters	Name	Type	Description
	<data>	Array	Structured array containing the appropriate data.
Return Value	None		

Python Syntax	AddTemperatureDataBlock(<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 Example	`oModule.AddTemperatureDataBlock(["data"])`

AddLibRefDataBlock

Use: Add a LibRef Data Block

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.
Return Value	None

Python Syntax	AddLibRefDataBlock(name data)
Python Example	`oModule.AddLibRefDataBlock("MyLibRefBlock" data)`

AddNetlistDataBlock

Use: Add a NetList Data Block

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.
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	AddNetlistDataBlock(data)
Python Example	`oModule.AddLibRefDataBlock(data)` *

AddPrintToAuditDataBlock

Adds a PrintToAudit data block.

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.
Return Value	None

Python Syntax	AddPrintToAuditDataBlock([<data>])
Python Example	`oModule.AddPrintToAuditDataBlock(["data"])`

AddStateVariableDataBlock

Adds a StateVariable data block.

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.
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	AddStateVariableDataBlock(<data>)
Python Example	`oModule.AddStateVariableDataBlock(["data"])`

AddSubstrateDataBlock

Adds a StateVariable data block.

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.
Return Value	None

Python Syntax	AddSubstrateDataBlock(<data>)
Python Example	`oModule.AddSubstrateDataBlock(["data"])`

AddDeviceNoiseDataBlock

Use: Add a DeviceNoise Data Block

UI Access	None
Parameters	Name | Type | Description <*data*> | Array | Structured array containing the appropriate data.

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	AddDeviceNoiseDataBlock()

Python Example	`oModule.AddDeviceNoiseDataBlock(`
	`  [`
	`  ]`

EditTemperatureDataBlock

Use: Edit a Temperature Data Block

UI Access	None		
Parameters	Name	Type	Description
	<name>	String	Temperature data block name
	<*data*>	Array	Structured array containing the appropriate data.
Return Value	None		

Python Syntax	EditTemperatureDataBlock(name data)
Python Example	`oModule.EditTemperatureDataBlock("MyTemperatureBlock" data)` *

EditLibRefDataBlock

Use: Edit a LibRef Data Block

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 <name> / String / LibRef data block name <*data*> / Array / Structured array containing the appropriate data.
Return Value	None

Python Syntax	EditLibRefDataBlock(name data)
Python Example	`oModule.EditLibRefDataBlock("MyLibRefBlock" data)`

EditNetlistDataBlock

Use: Edit a Netlist Data Block

UI Access	None
Parameters	Name / Type / Description <name> / String / Netlist data block name <*data*> / Array / Structured array containing the appropriate data.
Return Value	None

Python Syntax	EditNetlistDataBlock(name data)
Python Example	`oModule.EditNetlistDataBlock("MyNetlistBlock" data)` *

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditPrintToAuditDataBlock

Use: Edit a PrintToAudit Data Block

UI Access	None
Parameters	Name | Type | Description <name> | String | Print to audit data block name <*data*> | Array | Structured array containing the appropriate data.
Return Value	None

Python Syntax	EditPrintToAuditDataBlock(name data)
Python Example	`oModule.EditPrintToAuditDataBlock("PrintToAuditDataBlock" data)` *

EditStateVariableDataBlock

Use: Edit a StateVariable Data Block

UI Access	None
Parameters	Name | Type | Description <name> | String | State Variable data block name <*data*> | Array | Structured array containing the appropriate data.
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	EditStateVariableDataBlock(name data)
Python Example	`oModule.EditStateVariableDataBlock("MYStateVariableDataBlockName" data)` *

EditSubstrateDataBlock

Use: Edit a Substrate Data Block

UI Access	None
Parameters	Name | Type | Description <name> | String | Substrate data block name <*data*> | Array | Structured array containing the appropriate data.
Return Value	None

Python Syntax	EditSubstrateDataBlock(name data)
Python Example	`oModule.EditSubstrateDataBlock("MySubstrateBlock" data)` *

EditDeviceNoiseDataBlock

Use: Edit a DeviceNoise Data Block

UI Access	None
Parameters	Name | Type | Description <name> | String | Device noise data block name <*data*> | Array | Structured array containing the appropriate data.

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	EditDeviceNoiseDataBlock(name data)
Python Example	`oModule.EditDeviceNoiseDataBlock("MyDeviceNoiseDataBlock" data) *`

GetTemperatureDataBlock

Returns the TemperatureDataBlock name.

UI Access	Not available
Parameters	None
Return Value	Name of temperature data block

Python Syntax	GetTemperatureDataBlock()
Python Example	`tempStr = oModule.GetTemperatureDataBlock()`

GetAllLibRefDataBlocks

Use: Returns a name list of all library reference data blocks in the Nexxim Design

UI Access	None
Parameters	Name | Type | Description None
Return Value	Array of strings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetAllLibRefDataBlocks()
Python Example	`librefArr = oModule.GetAllLibRefDataBlocks()`

GetAllNetlistDataBlocks

Use: Returns a name list of all netlist data blocks in the Nexxim Design.

UI Access	None
Parameters	Name | Type | Description None
Return Value	Array of strings

Python Syntax	GetAllNetlistDataBlocks()
Python Example	`netlistArr = oModule.GetAllNetlistDataBlocks()`

GetAllSubstrateDataBlocks

Use: Returns a name list of all substrate data blocks in the Nexxim Design

UI Access	None
Parameters	Name | Type | Description None
Return Value	Array of strings

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Python Syntax	GetAllSubstrateDataBlocks()
Python Example	`subsArr = oModule.GetAllSubstrateDataBlocks()`

Remove [Nexxim]

Use: Removes a Data Block

UI Access	None
Parameters	Name | Type | Description <name> | string | Name of the data block to remove
Return Value	[none]

Python Syntax	Remove(<Data Block>)
Python Example	`oModule.Remove("MyDataBlock")` *

Rename [Nexxim]

Use: Renames a Data Block

UI Access	None
Parameters	Name | Type | Description <OldName> | string | Name of the data block to rename <NewName> | string | New desired name for data block
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	Rename(<OldName> <NewName>)
Python Example	`oModule.Rename("MyDataBlock" "SavedDataBlock")` *

Nexxim Simulation Setup Commands

The following sections present the commands that are available in the "SimSetup" module:

For example,

```
oDesign.GetModule("SimSetup")
oModule.CommandName <args>
```

Add

Add Alter Block [Nexxim]

Add AMI Analysis [Nexxim]

AddAnalysisOptions [Nexxim]

Add DC Analysis [Nexxim]

Add Envelope Setup [Nexxim]

Add HB 1-Tone [Nexxim]

Add HB N-Tone [Nexxim]

Add HSPICE Transient [Nexxim]

Add Linear Network Analysis [Nexxim]

Add Oscillator Analysis [Nexxim]

Add Oscillator N-Tone Analysis [Nexxim]

Add Oscillator Resonant Frequency Search [Nexxim]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddQuickEyeAnalysis
Add PXF Setup [Nexxim]
Add System FD Analysis [Nexxim]
Add Transient Analysis [Nexxim]
Add TV Noise [Nexxim]
AddVerifEyeAnalysis
AddSweep
Analyze
Analyze All Nominal [Nexxim]
AnalyzeSweep
Delete
DeleteSweep
DisplayBiasPointInfo
DynamicMeshOverlays
Edit
Edit Alter Block [Nexxim]
Edit AMI Analysis [Nexxim]
Edit Analysis Options Nexxim]
Edit DC Analysis [Nexxim]
Edit Envelope Setup[Nexxim]
Edit HB 1-Tone [Nexxim]

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit HB N-Tone [Nexxim]

Edit HSPICE Transient [Nexxim]

Edit Linear Network Analysis [Nexxim]

Edit Oscillator Analysis [Nexxim]

Edit Oscillator N-Tone Analysis [Nexxim]

Edit Oscillator Resonant Frequency Search [Nexxim]

Edit PXF Setup[Nexxim]

EditQuickEyeAnalysis

Edit System FD Analysis [Nexxim]

Edit TV Noise [Nexxim]

EditVerifEyeAnalysis

EditSweep

GetAllSolutionSetups

ListVariations

RenameSweep

RefreshMeshOverlays

Add [Nexxim]

Use: Adds a new setup

UI Access	

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 | | |
Return Value	[none]

Add Alter Block [Nexxim]

Use: Adds a new Alter Block

UI Access	
Parameters	Name | Type | Description alterdata | Array | Alter block data
Return Value	None

Python Syntax	AddAlterBlock(<data>[])

Python Example	`oModule.AddAlterBlock(`
	`  [`
	`  "NAME:AlterData",`
	`  "DataBlockID:=" , 22,`
	`  "Name:=" , "AlterData",`
	`  "Alter:=" ,`
	` ["Alter1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"V:=" , '''.ALTER first_one .param myval=2000 .HB TONES=2000 MAXk=3 .PRINT HB V(2)''']])

Add AMI Analysis [Nexxim]

Use: Adds a new AMI Analysis setup

UI Access	
Parameters	Name | Type | Description <newsetup> | Array | New set up data
Return Value	None

Python Syntax	AddAMIAnalysis(<setupData[]>)
Python Example	oModule.AddAMIAnalysis(["NAME:SimSetup", "DataBlockID:=" , 29,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "SimSetupID:=" , 1,
  "OptionName:=" , "(Default Options)",
  "AdditionalOptions:=" , "",
  "AlterBlockName:=" , "",
  "FilterText:=" , "",
  "AnalysisEnabled:=" , 1,
  [
    "NAME:OutputQuantities"
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "AMIAnalysis",
  "InputType:=" , 1,
  "AMIAnalysis:=" , [32,False]
])
```

Add DC Analysis [Nexxim]

Use: Adds a new DC analysis setup

UI Access	Right click Analysis > Add Nexxim Solution Setup > DC Analysis

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><setupData></td><td>Array</td><td>Array of attributes</td></tr></table>
Return Value	None

Python Syntax	AddDCAnalysis(<setupData[]>)

Python Example	<pre>oModule.AddDCAnalysis(
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 15,
	  "SimSetupID:=" , 3,
	  "OptionName:=" , "(Default Options)",
	  "AdditionalOptions:=" , "",
	  "AlterBlockName:=" , "",
	  "FilterText:=" , "",
	  "AnalysisEnabled:=" , 1,
	  [
	  "NAME:OutputQuantities"
	  ],</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:NoiseOutputQuantities"
    ],
    "Name:="                , "DCAnalysis1"
  ])
```

Add Envelope Setup [Nexxim]

Use: Adds a new Envelope Analysis setup

UI Access	Right click Analysis > Add Nexxim Solution Setup > Envelope Analysis
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><newsetup></td><td>Array</td><td>Array of attributes</td></tr></table>
Return Value	None

Python Syntax	AddEnvelopeSetup(<setupData[]>)

Python Example	
	oModule.AddEnvelopeSetup(
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 21,
	  "SimSetupID:=" , 26,
	  "OptionName:=" , "(Default Options)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "DC",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            ],
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ]
        ],
        [
          "NAME:NoiseOutputQuantities"
        ],
        "Name:="          , "Envelope",
        "EnvelopeData:="          , [1,"0","1e-8","","","trap","HB","No",0],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "EnvelopeMaxkValues:="    , ["F1","7"],
            "EnvelopeTonesValues:="   , ["1Ghz"],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
  ])
```

Add HB 1-Tone [Nexxim]

Use: Adds a new Harmonic Balance 1-Tone setup

UI Access	Right click Analysis > Add Nexxim Solution Setup > Harmonic Balance (1 Tone)
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><newsetup></td><td>array</td><td>Array of attributes</td></tr></table>
Return Value	None

Python Syntax	AddHB1Tone(<setupData[]>)

Python Example	`oModule.AddHB1Tone(`
	`[`
	`  "NAME:SimSetup",`
	`  "DataBlockID:="  , 9,`
	`  "SimSetupID:="  , 1,`
	`  "OptionName:="  , "(Default Options)",`
	`  "AdditionalOptions:="  , "",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AlterBlockName:="      , "",
"FilterText:="          , "",
"AnalysisEnabled:="     , 1,
[
  "NAME:OutputQuantities",
  [
    "NAME:Quantity",
    "NodeType:="          , "Harmonics",
    "CompID:="            , "",
    "CompName:="          , "",
    "QuantityName:="      , "2F1",
    "Selected:="          , True,
    "UnitType:="          , "NoUnit",
    "NetlistFUT:="                , "",
    "DataType:="          , "Real",
    "TypeInfo:="          , ["",0,"",
    "IsUnconstrained:="   , False],
    "CircuitInstanceID:=" , ""
  ],
  [
    "NAME:Quantity",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "3F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="              , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "4F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "DC",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:Quantity",
       "NodeType:="          , "Harmonics",
       "CompID:="            , "",
       "CompName:="          , "",
       "QuantityName:="      , "F1",
       "Selected:="          , True,
       "UnitType:="          , "NoUnit",
       "NetlistFUT:="        , "",
       "DataType:="          , "Real",
       "TypeInfo:="          , ["",0,"",
       "IsUnconstrained:="   , False],
       "CircuitInstanceID:=" , ""
     ],
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:="                , "HB1Tone",
  "HB1ToneData:="         , [7,"1GHz","HB","No","0.0"],
  "HarmonicsText:="       , "2F1 3F1 4F1 DC F1"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Add HB N-Tone [Nexxim]

Use: Adds a new Harmonic Balance N-Tone setup

UI Access	Right click Analysis > Add Nexxim Solution Setup > Harmonic Balance (N Tone)
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><newsetup></td><td>array</td><td>Array of attributes</td></tr></table>
Return Value	None

Python Syntax	AddHBnTone(<setupData[]>)
Python Example	oModule.AddHBnTone(["NAME:SimSetup", "DataBlockID:=" , 17, "SimSetupID:=" , 8, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "", "AnalysisEnabled:=" , 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:OutputQuantities",
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "2F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="                , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "QuantityName:="      , "3F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="              , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "4F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "DC",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "CompName:="          , "",
        "QuantityName:="      , "F1",
        "Selected:="          , True,
        "UnitType:="          , "NoUnit",
        "NetlistFUT:="        , "",
        "DataType:="          , "Real",
        "TypeInfo:="          , ["",0,"",
        "IsUnconstrained:="   , False],
        "CircuitInstanceID:=" , ""
      ],
    ],
    [
      "NAME:NoiseOutputQuantities"
    ],
    "Name:="                , "HBnTone",
    "HB1ToneData:="         , [1,"HB","No","0.0", "0.0","F1","7"],
    "HBnValues:="           , ["1GHz"]
    "HarmonicsText:="       , "2F1 3F1 4F1 DC F1"
])
```

Add HSPICE Transient [Nexxim]

Use: Adds a new HSPICE Transient Analysis 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>Right click Analysis > Add HSPICE Solution Setup > Transient Analysis</td></tr>
<tr><td>Parameters</td><td>
<table>
<tr><td>Name</td><td>Type</td><td>Description</td></tr>
<tr><td><setupData></td><td>array</td><td>Array of attributes</td></tr>
</table>
</td></tr>
<tr><td>Return Value</td><td>None</td></tr>
</table>

Python Syntax	AddHSPICETransient(<setupData[]>)

Python Example	<pre>oModule.AddHSPICETransient(
	[
	  "NAME:SimSetup",
	  "DataBlockID:=" , 30,
	  "SimSetupID:=" , 9,
	  "OptionName:=" , "(Default Options)",
	  "AdditionalOptions:=" , "",
	  "AlterBlockName:=" , "",
	  "FilterText:=" , "",
	  "AnalysisEnabled:=" , 1,
	  [
	  "NAME:OutputQuantities"</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:NoiseOutputQuantities"
  ],
  "Name:=" , "HSPICETransient",
  "HSPICETransientData:=" , ["0.1ns","10ns"],
  "HSPICETransientOtherData:=", [3]
])
```

Add Linear Network Analysis [Nexxim]

Use: Adds a new Linear Network analysis setup

UI Access	None
Parameters	Name | Type | Description <setupData> | Array | Array of attributes
Return Value	[none]

Python Syntax	AddLinearNetworkAnalysis(<setupData[]>)
Python Example	`oModule.AddLinearNetworkAnalysis(` `  [`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"NAME:SimSetup",
"DataBlockID:="         , 16,
"SimSetupID:="          , 15,
"OptionName:="          , "(Default Options)",
"AdditionalOptions:="   , "",
"AlterBlockName:="      , "",
"FilterText:="          , "",
"AnalysisEnabled:="     , 1,
[
        "NAME:OutputQuantities"
],
[
        "NAME:NoiseOutputQuantities"
],
"Name:="                , "LinearFrequency",
"LinearFrequencyData:=" , [False,"0.1",False,"",False],
[
        "NAME:SweepDefinition",
        "Variable:="    , "F",
        "Data:="        , "LINC 1GHz 100GHz 10",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "OffsetF1:="            , False,
            "Synchronize:="         , 0
    ]
  ])
```

Add Oscillator Analysis [Nexxim]

Use: Adds a new Oscillator 1-Tone analysis setup

UI Access	
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><setupData[]></td><td>Array</td><td>Array of attributes</td></tr></table>
Return Value	[none]

Python Syntax	AddOscillatorAnalysis(<setupData[]>)
Python Example	

```
oModule.AddOscillatorAnalysis(
  [
    "NAME:SimSetup",
    "DataBlockID:="         , 21,
    "SimSetupID:="          , 26,
    "OptionName:="          , "(Default Options)",
    "AdditionalOptions:="   , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AlterBlockName:="      , "",
"FilterText:="          , "",
"AnalysisEnabled:="     , 1,
[
        "NAME:OutputQuantities",
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "2F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "DC",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ]
        ],
        [
          "NAME:NoiseOutputQuantities"
        ],
        "Name:="          , "Oscillator",
        "OscillatorData:="        , [7,"1Ghz",False,1],
        "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Add Oscillator N-Tone Analysis [Nexxim]

Use: Adds a new Oscillator N-Tone analysis setup

UI Access	
Parameters	<table><tr><td>Name</td><td>Type</td><td>Description</td></tr><tr><td><setupData[]></td><td>Array</td><td>Array of attributes</td></tr></table>
Return Value	[none]

Python Syntax	AddOscillatorNtoneAnalysis(<setupData[]>)
Python Example	oModule.AddOscillatorNtoneAnalysis(["NAME:SimSetup", "DataBlockID:=" , 23, "SimSetupID:=" , 1, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AnalysisEnabled:=" , 1,
[
"NAME:OutputQuantities",
 [
 "NAME:Quantity",
 "NodeType:=" , "Harmonics",
 "CompID:=" , "",
 "CompName:=" , "",
 "QuantityName:=" , "2F2",
 "Selected:=" , True,
 "UnitType:=" , "NoUnit",
 "NetlistFUT:=" , "",
 "DataType:=" , "Real",
 "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
 "CircuitInstanceID:=" , ""
],
[
 "NAME:Quantity",
 "NodeType:=" , "Harmonics",
 "CompID:=" , "",
 "CompName:=" , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
 "QuantityName:=" , "3F2",
 "Selected:=" , True,
 "UnitType:=" , "NoUnit",
 "NetlistFUT:=" , "",
 "DataType:=" , "Real",
 "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
 "CircuitInstanceID:=" , ""
],
[
 "NAME:Quantity",
 "NodeType:=" , "Harmonics",
 "CompID:=" , "",
 "CompName:=" , "",
 "QuantityName:=" , "4F2",
 "Selected:=" , True,
 "UnitType:=" , "NoUnit",
 "NetlistFUT:=" , "",
 "DataType:=" , "Real",
 "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
 "CircuitInstanceID:=" , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
     "NAME:Quantity",
     "NodeType:=" , "Harmonics",
     "CompID:=" , "",
     "CompName:=" , "",
     "QuantityName:=" , "DC",
     "Selected:=" , True,
     "UnitType:=" , "NoUnit",
     "NetlistFUT:=" , "",
     "DataType:=" , "Real",
     "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
     "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
     "NodeType:=" , "Harmonics",
     "CompID:=" , "",
     "CompName:=" , "",
     "QuantityName:=" , "F2",
     "Selected:=" , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "UnitType:=" , "NoUnit",
      "NetlistFUT:=" , "",
      "DataType:=" , "Real",
      "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
      "CircuitInstanceID:=" , ""
    ],
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "OscillatorNtone",
  "OscillatorNtoneData:=" , [1,1,False,1],
  "OscMaxkValues:=" , ["F1","7"],
  "OscTonesValues:=" , ["1Ghz"],
  "DrivenMaxkValues:=" , ["RF1","7"],
  "DrivenTonesValues:=" , ["1Ghz"],
  "HarmonicsText:=" , "2F2 3F2 4F2 DC F2",
    [
    "NAME:SweepDefinition",
    "Variable:=" , "FNoi",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "Data:=" , "LINC 1GHz 200GHz 10",
    "OffsetF1:=" , False,
    "Synchronize:=" , 0
    ]
  ])
```

Add Oscillator Resonant Frequency Search [Nexxim]

Use: Adds a new Oscillator Resonant Frequency Search setup

UI Access	
Parameters	Name | Type | Description <setupData[]> | Array | Array of attributes
Return Value	[none]

Python Syntax	AddOscResonantFreqencySearch(<setupData[]>)
Python Example	

```
oModule.AddOscResonantFrequencySearch(
  [
    "NAME:SimSetup",
    "DataBlockID:="         , 21,
    "SimSetupID:="          , 26,
    "OptionName:="          , "(Default Options)",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AdditionalOptions:="   , "",
"AlterBlockName:="      , "",
"FilterText:="          , "",
"AnalysisEnabled:="     , 1,
[
        "NAME:OutputQuantities",
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "2F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "DC",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ]
    ],
    [
      "NAME:NoiseOutputQuantities"
    ],
    "Name:="          , "OscResonantFrequency",
    "OscResonantFrequencyData:="      ,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	[1000,"1000000000:,"5000000000","1Ghz",False,1], "OscMaxkValues:=" , ["F1","7" "OscToneValues:=" , ["1Ghz"]])

Add PXF Setup [Nexxim]

Use: Adds a new Periodic Transfer Function (PXF) setup

Syntax: AddPXFSetup(

ARRAY newsetup) // new setup data

Python Syntax	AddPXFSetup(<setupData[]>)
Python Example	oModule.AddPXFSetup(["NAME:SimSetup", "DataBlockID:=" , 21, "SimSetupID:=" , 26, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AnalysisEnabled:="     , 1,
[
        "NAME:OutputQuantities",
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "2F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "CompName:="            , "",
                    "QuantityName:="        , "3F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "4F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "DC",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ]
            ],
            [
              "NAME:NoiseOutputQuantities"
            ],
            "Name:="          , "PXF",
            "PXFData:="               , [1,"HB",False,"HB","No",0],
            "PXFMaxkValues:=" , ["F1","7"],
            "PXFTonesValues:="        , ["1Ghz"],
            "PXFRelHarmNums:="        , [],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "PXFMaxSideband:="        , ["0"],
            "PXFSideband:="           , ["0"],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
            [
               "NAME:SweepDefinition",
               "Variable:="           , "F",
               "Data:="               , "LINC 10GHz 100GHz 10",
               "OffsetF1:="           , False,
               "Synchronize:="                , 0
            ],
            [
               "NAME:SweepDefinition",
               "Variable:="           , "F",
               "Data:="               , "LINC 10GHz 100GHz 10",
               "OffsetF1:="           , False,
               "Synchronize:="                , 0
            ],
  ])
```

AddQuickEyeAnalysis

Adds a QuickEye analysis to the design.

Command: Context menu of Analysis icon in project tree -> AddSolution Setup... -> 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.

Syntax: AddQuickEyeAnalysis (Array("NAME:SimSetupName",

```
"DataBlockID:=", <int>,
"SimSetupID:=", <int>,
"OptionName:=", <string>,
"AdditionalOptions:=", <string>,
"AlterBlockName:=", <string>,
"FilterText:=", <string>,
"AnalysisEnabled:=", <int>, // 1 if enabled, 0 if disabled
Array("NAME:OutputQuantities", <QuantityArray>, <QuantityArray>...)
Array("NAME:NoiseOutputQuantities", <QuantityArray>, <QuantityArray>...) ),
"Name:=", <string>, // name for new analysis
"QuickEyeAnalysis:=", Array(<string>, // Input rise time
<string>, // Input low voltage
<string>, // Input high voltage
<string>, // bits per second
<string>, // number of FFE taps
<string>, // Random Jitter Standard Deviation
<string>, // Delay
<string>, // Duty cycle distortion
<bool>, // true if unit interval, false if bits per second
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
<string>, // number of DFE taps
<bool>, // true to calculate FFE, false if weights are specified
<bool>, // true to calculate DFE, false if weights are specified
<string>, // DFE decision threshold
<string>, // DFE decision high
<string>, // DFE decision low
<bool>), // true if using specified equalization, false if disabled
Array("NAME:SweepDefinition",
"Variable:=", <string>,
"Data:=", <string>, // sweep
"OffsetF1:=", <bool>,
"Synchronize:=", <int>), // 1 to Synchronize, 0 otherwise
"FFEWts:=", Array(".5", "-2"), // optional, specified weights
"DFEWts:=", Array("2"))) // optional, specified weights
```

Return Value: <string> – // Name of the analysis that was added

// If the name requested conflicts with the name of an existing

// analysis, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: Parameters: <QuantityArray>:

Array("NAME:Quantity",

"NodeType:=", <string>, // CompInst, Variable, Net, Harmonics, or Custom

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"CompID:=", <string>,

"CompName:=", <string>,

"QuantityName:=", <string>,

"Selected:=", <bool>,

"UnitType:=", <string>,

"DataType:=", <string>, // Real, Complex, Integer, Enum, Char, Free, Array, Record

"CircuitInstanceID:=", <string)

Python Syntax	AddQuickEyeAnalysis(<data_block>)
Python Example	name = oModule.AddQuickEyeAnalysis (["NAME:SimSetup", "DataBlockID:=", 28, "SimSetupID:=", 3, "OptionName:=", "(Default Options)", "AdditionalOptions:=","", "AlterBlockName:=", "", "FilterText:=", "", "AnalysisEnabled:=",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:OutputQuantities",
           [
              "NAME:Quantity",
              "NodeType:=","CompInst",
              "CompID:=", "5",
              "CompName:=", "Level01_NPN_Model_5",
              "QuantityName:=", "I",
              "Selected:=", True,
              "UnitType:=", "NoUnit",
              "DataType:=","Real",
              "CircuitInstanceID:=", ""
           ],
           [
              "NAME:Quantity",
              "NodeType:=", "CompInst",
              "CompID:=", "10",
              "CompName:=", "RES__10",
              "QuantityName:=", "I",
              "Selected:=", True,
              "UnitType:=", "NoUnit",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ],
         [
            "NAME:Quantity",
            "NodeType:=", Net",
            "CompID:=", "",
            "CompName:=", "net_47",
            "QuantityName:=", "V",
            "Selected:=", True,
            "UnitType:=", "NoUnit",
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ]
      ],
      [
         "NAME:NoiseOutputQuantities"
      ],
     "Name:=", "MyQuickEyeAnalysis",
     "QuickEyeAnalysis:=",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      [
         "5e-10", "0", "1", "1e-9", "2", "0", "1e-9", "0",
         True, "1", False, False,"0", "1", "-1", True
      ],
      [
         "NAME:SweepDefinition",
         "Variable:=", "Temp",
         "Data:=", "LINC 1cel 10cel 10",
         "OffsetF1:=", False,
         "Synchronize:=", 0),
         "FFEWts:=",
         [
            ".5", "-2"
         ]
         DFEWts:=",
         [
            "2"
         ]
      ]
   ]
)
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Add System FD Analysis [Nexxim]

Use: Adds a new System Frequency Domain (FD) Analysis setup

Syntax: AddSystemFDAnalysis(

```
ARRAY newsetup) // new setup data
```

Python Syntax	AddSystemFDAnalysis(<setupData[]>)
Python Example	oModule.AddSystemFDAnalysis(["NAME:SimSetup", "DataBlockID:=" , 32, "SimSetupID:=" , 16, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "", "AnalysisEnabled:=" , 1, ["NAME:OutputQuantities"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "SystemFDAnalysis",
  "SystemFDAnalysis:=" , [False]
])
```

Add Transient Analysis [Nexxim]

Use: Adds a new Transient analysis setup

Syntax: AddTransient(

ARRAY newsetup) // new setup data

Python Syntax	AddTransient(<setupData[]>)

Python Example	oModule.AddTransient(
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 10,
	  "SimSetupID:=" , 11,
	  "OptionName:=" , "(Default Options)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities"
    ],
    [
            "NAME:NoiseOutputQuantities"
    ],
    "Name:="                , "NexximTransient1",
    "TransientData:="       , ["0.1ns","10ns"],
    "TransientNoiseData:="  , [False,"","",0,1,0,False,1],
    "TransientOtherData:="  , ["default"]
  ])
```

Add TV Noise [Nexxim]

Use: Adds a new TV Noise analysis setup

Syntax: AddTVNoise(

ARRAY newsetup) // new setup 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	AddTVNoise(<setupData[]>)
Python Example	oModule.AddTVNoise(["NAME:SimSetup", "DataBlockID:=" , 20, "SimSetupID:=" , 21, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "", "AnalysisEnabled:=" , 1, ["NAME:OutputQuantities", ["NAME:Quantity", "NodeType:=" , "Harmonics", "CompID:=" , "", "CompName:=" , "", "QuantityName:=" , "2F1", "Selected:=" , True,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "3F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "DC",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "CircuitInstanceID:="   , ""
            ]
        ],
        [
            "NAME:NoiseOutputQuantities"
        ],
        "Name:="          , "TVNoise",
        "TVNoiseData:="           , [1,"","","","","HB",False,"HB","No",0],
        "TVNoiseMaxkValues:="     , ["F1","7"],
        "TVNoiseTonesValues:="    , ["1Ghz"],
        "TVNoiseRelHarmNums:="    , [],
        "TVNoiseRefSideband:="    , ["1"],
        "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
        [
            "NAME:SweepDefinition",
            "Variable:="          , "F",
            "Data:="              , "LINC 10GHz 100GHz 10",
            "OffsetF1:="          , False,
            "Synchronize:="       , 0
        ],
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "NAME:FSweep",
            "Variable:="            , "F",
            "Data:="                , "LINC 10GHz 100GHz 10",
            "OffsetF1:="            , False,
            "Synchronize:="         , 0
          ]
  ])
```

Add VerifEye Analysis

Adds a VerifEye analysis to the design.

Command: Context menu of Analysis icon in project tree -> AddSolution Setup... -> VerifEye (Statistical Eye) Analysis

Syntax: AddVerifEyeAnalysis (Array("NAME:SimSetupName",

"DataBlockID:=", <int>,

"SimSetupID:=", <int>,

"OptionName:=", <string>,

"AdditionalOptions:=", <string>,

"AlterBlockName:=", <string>,

"FilterText:=", <string>,

"AnalysisEnabled:=", <int>, // 1 if enabled, 0 if disabled

Array("NAME:OutputQuantities", <QuantityArray>, <QuantityArray>...)

Array("NAME:NoiseOutputQuantities", <QuantityArray>, <QuantityArray>...)),

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>, // name for new analysis

"VerifEyeAnalysis:=", Array(<string>, // Input rise time

<string>, // Input low voltage

<string>, // Input high voltage

<string>, // bits per second

<string>, // number of FFE taps

<string>, // Random Jitter Standard Deviation

<string>, // Delay

<string>, // Duty cycle distortion

<bool>, // true if unit interval, false if bits per second

<string>, // number of DFE taps

<bool>, // true to calculate FFE, false if weights are specified

<bool>, // true to calculate DFE, false if weights are specified

<string>, // DFE decision threshold

<string>, // DFE decision high

<string>, // DFE decision low

<bool>), // true if using specified equalization, false if disabled

Array("NAME:SweepDefinition",

"Variable:=", <string>,

"Data:=", <string>, // sweep

"OffsetF1:=", <bool>,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

"Synchronize:=", <int>), // 1 to Synchronize, 0 otherwise

"FFEWts:=", Array(".5", "-2"), // optional, specified weights

"DFEWts:=", Array("2"))) // optional, specified weights

Return Value: <string> – // Name of the analysis that was added

// If the name requested conflicts with the name of an existing

// analysis, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <QuantityArray>:

Array("NAME:Quantity",

"NodeType:=", <string>, // CompInst, Variable, Net, Harmonics, or Custom

"CompID:=", <string>,

"CompName:=", <string>,

"QuantityName:=", <string>,

"Selected:=", <bool>,

"UnitType:=", <string>,

"DataType:=", <string>, // Real, Complex, Integer, Enum, Char, Free, Array, Record

"CircuitInstanceID:=", <string)

Python Syntax	AddVerifEyeAnalysis(<data_block>)
Python Example	`name = oModule.AddVerifEyeAnalysis (` [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"NAME:SimSetup",
"DataBlockID:=",
28,
"SimSetupID:=", 3,
"OptionName:=", "(Default Options)",
"AdditionalOptions:=","",
"AlterBlockName:=", "",
"FilterText:=", "",
"AnalysisEnabled:=",1,
   [
      "NAME:OutputQuantities",
      [
         "NAME:Quantity",
         "NodeType:=","CompInst",
         "CompID:=", "5",
         "CompName:=", "Level01_NPN_Model_5",
         "QuantityName:=", "I",
         "Selected:=", True,
         "UnitType:=", "NoUnit",
         "DataType:=","Real",
         "CircuitInstanceID:=", ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
           "NAME:Quantity",
           "NodeType:=", "CompInst",
           "CompID:=", "10",
           "CompName:=", "RES__10",
           "QuantityName:=", "I",
           "Selected:=", True,
           "UnitType:=", "NoUnit",
           "DataType:=", "Real",
           "CircuitInstanceID:=", ""
        ],
        [
           "NAME:Quantity",
           "NodeType:=", Net",
           "CompID:=", "",
           "CompName:=", "net_47",
           "QuantityName:=", "V",
           "Selected:=", True,
           "UnitType:=", "NoUnit",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ]
      ],
      [
         "NAME:NoiseOutputQuantities"
      ],
     "Name:=", "MyVerifEyeAnalysis",
     "VerifEyeAnalysis:=",
      [
         "5e-10", "0", "1", "1e-9", "2", "0", "1e-9", "0",
         True, "1", False, False,"0", "1", "-1", True
      ],
      [
         "NAME:SweepDefinition",
         "Variable:=", "Temp",
         "Data:=", "LINC 1cel 10cel 10",
         "OffsetF1:=", False,
         "Synchronize:=", 0),
         "FFEWts:=",
         [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            ".5", "-2"
         ]
         DFEWts:=",
         [
            "2"
         ]
      ]
   ]
)
```

AddSweep [Nexxim]

Use: Adds a sweep to a setup

Syntax: AddSweep(

```
STRING setup) // setup name
STRING newsweep) // new sweep name
```

Analyze All Nominal [Nexxim]

Simulates all defined setups

Syntax: AnalyzeAllNominal( )

Ansys 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()`

Analyze [Nexxim]

Simulates the given setup

Syntax: Analyze(STRING setup1) // setup name

Python Syntax	Analyze(<setupName>)
Python Example	`oDesign.Analyze("QuickEyeAnalysis")`

AnalyzeSweep [Nexxim]

Use: Simulates the given sweep

Syntax: AnalyzeSweep(

```
STRING setup1 // setup name
STRING sweep1) // sweep name
```

Delete [Nexxim]

Use: Removes a simulation setup

Syntax: Delete(

```
STRING name) // The specified setup to delete
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DisplayBiasPointInfo

Use: Display voltage/current bias

Command: None

Syntax: DisplayBiasPointInfo <display_preference>

Return Value: None

Parameters: `<display_preference>`

Type: string

Python Syntax	DisplayBiasPointInfo(<display_preference>)
Python Example	`oDesign.DisplayBiasPointInfo(3)`

DeleteSweep [Nexxim]

Use: Removes a sweep from a setup

Syntax: DeleteSweep(

```
STRING setup1) // The specified setup
STRING sweep1) // The specified sweep
```

DynamicMeshOverlays

Use: Turns on dynamic mesh overlays

Syntax: DynamicMeshOverlays(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
STRING setup) // setup name
```

Edit [Nexxim]

Use: Edit a setup

Syntax: Edit(

```
STRING setup) // setup name
ARRAY solvesetup_data) // setup data
```

Edit Alter Block [Nexxim]

Use: Edits an Alter Block

Syntax: EditAlterBlock(

```
STRING alterblockname,
ARRAY alterdata) // Alter block data
```

Python Syntax	EditAlterBlock(<name>,<data>[])

Python Example	`oModule.EditAlterBlock("AlterData",`
	`  [`
	`  "NAME:AlterData",`
	`  "DataBlockID:=" , 22,`
	`  "Name:=" , "AlterData",`
	`  "Alter:=" ,`
	`  ["Alter1",`
	`  "V:=" , '''.ALTER first_one`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	.param myval=2000 .HB TONES=2000 MAXk=5 .PRINT HB V(2)''']])

Edit AMI Analysis [Nexxim]

Use: Edits an AMI Analysis setup

Syntax: EditAMIAnalysis(

```
STRING name,
ARRAY newsetup) // new setup data
```

Python Syntax	AddAMIAnalysis(<name>,<setupData[]>)
Python Example	oModule.AddAMIAnalysis("AMIAnalysis1", ["NAME:SimSetup", "DataBlockID:=" , 29, "SimSetupID:=" , 1, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "AlterBlockName:=" , "",
  "FilterText:=" , "",
  "AnalysisEnabled:=" , 1,
  [
    "NAME:OutputQuantities"
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "AMIAnalysis",
  "InputType:=" , 1,
  "AMIAnalysis:=" , [32,False]
  [
    "NAME:SweepDefinition",
    "Variable:=" , "Temp",
    "Data:=" , "LINC 30cel 300cel 10",
    "OffsetF1:=" , False,
    "Synchronize:=" , 0
  ]
])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit DC Analysis [Nexxim]

Use: Edits a DC analysis setup

Syntax: EditDCAnalysis(

```
ARRAY setup) // modified setup data
```

Python Syntax	EditDCAnalysis(<modifiedsetupData[]>)

Python Example	oModule.AddDCAnalysis(
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 15,
	  "SimSetupID:=" , 3,
	  "OptionName:=" , "(Default Options)",
	  "AdditionalOptions:=" , "",
	  "AlterBlockName:=" , "",
	  "FilterText:=" , "",
	  "AnalysisEnabled:=" , 1,
	  [
	  "NAME:OutputQuantities"

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
            "NAME:NoiseOutputQuantities"
    ],
    "Name:="                , "DCAnalysis1"
    [
            "NAME:SweepDefinition",
            "Variable:="            , "Temp",
            "Data:="                , "LINC 10cel 200cel 10 LINC 20cel 400cel 10",
            "OffsetF1:="            , False,
            "Synchronize:="         , 0
    ]
  ])
```

Edit Envelope Setup[Nexxim]

Use: Edits an Envelope Analysis setup

Syntax: EditEnvelopeSetup(

```
STRING name,
ARRAY newsetup) // new setup data
```

Python Syntax	EditEnvelopeSetup(<name>,<setupData[]>)
Python Example	

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oModule.EditEnvelopeSetup("Envelope",
  [
    "NAME:SimSetup",
    "DataBlockID:="         , 21,
    "SimSetupID:="          , 26,
    "OptionName:="          , "(Default Options)",
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "4F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ],
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "DC",
                    "Selected:="            , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ]
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            ],
            [
              "NAME:NoiseOutputQuantities"
            ],
            "Name:="          , "PXF",
            "PXFData:="       , [1,"HB",False,"HB","No",0],
            "PXFMaxkValues:=" , ["F1","7"],
            "PXFTonesValues:="        , ["1Ghz"],
            "PXFRelHarmNums:="        , [],
            "PXFMaxSideband:="        , ["0"],
            "PXFSideband:="           , ["0"],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
            [
               "NAME:SweepDefinition",
               "Variable:="           , "F",
               "Data:="               , "LINC 10GHz 100GHz 10",
               "OffsetF1:="           , False,
               "Synchronize:="                , 0
            ],
            "Name:="          , "Envelope",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"EnvelopeData:=" , [1,"0","1e-8","","","trap","HB","No",0], "EnvelopeMaxkValues:=" , ["F1","7"], "EnvelopeTonesValues:=" , ["1Ghz"], "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",])

Edit HB 1-Tone [Nexxim]

Use: Edits a Harmonic Balance 1-Tone setup

Syntax: EditHB1Tone(

STRING setupname

ARRAY setup) // setup data

Python Syntax	EditHB1Tone(<setupName>,<setupData[]>)
Python Example	oModule.EditHB1Tone("HB1Tone", ["NAME:SimSetup", "DataBlockID:=" , 9, "SimSetupID:=" , 1, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "", "AnalysisEnabled:=" , 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:OutputQuantities",
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "2F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="              , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "CompName:="          , "",
      "QuantityName:="      , "3F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="                , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "4F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "DC",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:="               , "HB1Tone",
  "HB1ToneData:="        , [7,"1GHz","HB","No","0.0"],
  "HarmonicsText:="      , "2F1 3F1 4F1 DC F1"
  [
    "NAME:SweepDefinition",
    "Variable:="           , "Temp",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Data:=" , "LINC 10cel 200cel 10", "OffsetF1:=" , False, "Synchronize:=" , 0]])

Edit HB N-Tone [Nexxim]

Use: Edits a Harmonic Balance N-Tone setup

Syntax: EditHBnTone(

STRING setupname

ARRAY setup) // setup data

Python Syntax	EditHBnTone(<setupName>,<setupData[]>)
Python Example	oModule.EditHBnTone("HBnTone", ["NAME:SimSetup", "DataBlockID:=" , 17, "SimSetupID:=" , 8, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"FilterText:="          , "",
"AnalysisEnabled:="     , 1,
[
  "NAME:OutputQuantities",
  [
    "NAME:Quantity",
    "NodeType:="          , "Harmonics",
    "CompID:="            , "",
    "CompName:="          , "",
    "QuantityName:="      , "2F1",
    "Selected:="          , True,
    "UnitType:="          , "NoUnit",
    "NetlistFUT:="                , "",
    "DataType:="          , "Real",
    "TypeInfo:="          , ["",0,"",
    "IsUnconstrained:="   , False],
    "CircuitInstanceID:=" , ""
  ],
  [
    "NAME:Quantity",
    "NodeType:="          , "Harmonics",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "3F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="              , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
      "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "4F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "DC",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NodeType:="          , "Harmonics",
      "CompID:="            , "",
      "CompName:="          , "",
      "QuantityName:="      , "F1",
      "Selected:="          , True,
      "UnitType:="          , "NoUnit",
      "NetlistFUT:="        , "",
      "DataType:="          , "Real",
      "TypeInfo:="          , ["",0,"",
      "IsUnconstrained:="   , False],
      "CircuitInstanceID:=" , ""
    ],
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:="                , "HBnTone",
  "HBnTone:="             , [7,"HB","No","0.0","0.0","F1","7"],
  "HBnValues:="           , [1GHz],
  "HarmonicsText:="       , "2F1 3F1 4F1 DC F1"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	["NAME:SweepDefinition", "Variable:=" , "Temp", "Data:=" , "LINC 10cel 200cel 10", "OffsetF1:=" , False, "Synchronize:=" , 0]])

Edit HSPICE Transient [Nexxim]

Use: Edits an HSPICE Transient Analysis setup

Syntax: EditHSPICETransient(

STRING setupname,

ARRAY newsetup) // new setup data

Python Syntax	EditHSPICETransient(<name>,<setupData[]>)
Python Example	oModule.AddHSPICETransient("HSPICETransient", ["NAME:SimSetup", "DataBlockID:=" , 30, "SimSetupID:=" , 9, "OptionName:=" , "(Default Options)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
  "AdditionalOptions:=" , "",
  "AlterBlockName:=" , "",
  "FilterText:=" , "",
  "AnalysisEnabled:=" , 1,
  [
    "NAME:OutputQuantities"
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "HSPICETransient",
  "HSPICETransientData:=" , ["0.1ns","10ns"],
  "HSPICETransientOtherData:=", [3]
])
```

Edit Linear Network Analysis [Nexxim]

Use: Edits a Linear Network analysis setup

Syntax: EditLinearNetworkAnalysis(

```
STRING setupname
ARRAY newsetup) // modified setup 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	EditLinearNetworkAnalysis(<setupName>,<setupData[]>)
Python Example	oModule.EditLinearNetworkAnalysis("LinearFrequency1", ["NAME:SimSetup", "DataBlockID:=" , 16, "SimSetupID:=" , 17, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "", "AnalysisEnabled:=" , 1, ["NAME:OutputQuantities"], ["NAME:NoiseOutputQuantities"], "Name:=" , "LinearFrequency", "LinearFrequencyData:=" , [True,"0.1",False,"",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:SweepDefinition", "Variable:=" , "F", "Data:=" , "LINC 1GHz 100GHz 10", "OffsetF1:=" , False, "Synchronize:=" , 0]])

Edit Oscillator Analysis [Nexxim]

Use: Edits an Oscillator analysis setup

Syntax: EditOscillatorAnalysis(

STRING setup

ARRAY setupdata) // modified setup data

Python Syntax	EditOscillatorAnalysis(<setupName>,<setupData[]>)
Python Example	oModule.EditOscillatorAnalysis("Oscillator", ["NAME:SimSetup", "DataBlockID:=" , 21, "SimSetupID:=" , 26,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "OptionName:="          , "(Default Options)",
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "DC",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ]
    ],
    [
      "NAME:NoiseOutputQuantities"
    ],
    "Name:="          , "Oscillator",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "OscillatorData:="        , [7,"1Ghz",False,1],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
            [
              "NAME:SweepDefinition",
              "Variable:="            , "Fnoi",
              "Data:="                , "LINC 10GHz 100GHz 10",
              "OffsetF1:="            , False,
              "Synchronize:="         , 0
            ],
  ])
```

Edit Oscillator N-Tone Analysis [Nexxim]

Use: Edits an Oscillator N-Tone analysis setup

Syntax: EditOscillatorNtoneAnalysis(

```
STRING setupname,
ARRAY newsetup) // new setup data
```

Python Syntax	EditOscillatorNtoneAnalysis(<name>,<setupData[]>)

Python Example	oModule.EditOscillatorNtoneAnalysis("OscillatorNtone",
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 23,

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"SimSetupID:=" , 1,
"OptionName:=" , "(Default Options)",
"AdditionalOptions:=" , "",
"AlterBlockName:=" , "",
"FilterText:=" , "",
"AnalysisEnabled:=" , 1,
[
"NAME:OutputQuantities",
 [
 "NAME:Quantity",
 "NodeType:=" , "Harmonics",
 "CompID:=" , "",
 "CompName:=" , "",
 "QuantityName:=" , "2F2",
 "Selected:=" , True,
 "UnitType:=" , "NoUnit",
 "NetlistFUT:=" , "",
 "DataType:=" , "Real",
 "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
 "CircuitInstanceID:=" , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    ],
    [
     "NAME:Quantity",
     "NodeType:=" , "Harmonics",
     "CompID:=" , "",
     "CompName:=" , "",
     "QuantityName:=" , "3F2",
     "Selected:=" , True,
     "UnitType:=" , "NoUnit",
     "NetlistFUT:=" , "",
     "DataType:=" , "Real",
     "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
     "CircuitInstanceID:=" , ""
    ],
    [
     "NAME:Quantity",
     "NodeType:=" , "Harmonics",
     "CompID:=" , "",
     "CompName:=" , "",
     "QuantityName:=" , "4F2",
     "Selected:=" , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "UnitType:=" , "NoUnit",
        "NetlistFUT:=" , "",
        "DataType:=" , "Real",
        "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
        "CircuitInstanceID:=" , ""
       ],
       [
        "NAME:Quantity",
        "NodeType:=" , "Harmonics",
        "CompID:=" , "",
        "CompName:=" , "",
        "QuantityName:=" , "DC",
        "Selected:=" , True,
        "UnitType:=" , "NoUnit",
        "NetlistFUT:=" , "",
        "DataType:=" , "Real",
        "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
        "CircuitInstanceID:=" , ""
       ],
       [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
      "NAME:Quantity",
      "NodeType:=" , "Harmonics",
      "CompID:=" , "",
      "CompName:=" , "",
      "QuantityName:=" , "F2",
      "Selected:=" , True,
      "UnitType:=" , "NoUnit",
      "NetlistFUT:=" , "",
      "DataType:=" , "Real",
      "TypeInfo:=" , ["",0,"", "IsUnconstrained:=" , False],
      "CircuitInstanceID:=" , ""
    ],
  ],
  [
    "NAME:NoiseOutputQuantities"
  ],
  "Name:=" , "OscillatorNtone",
  "OscillatorNtoneData:=" , [1,1,True,1],
  "OscMaxkValues:=" , ["F1","7"],
  "OscTonesValues:=" , ["1Ghz"],
  "DrivenMaxkValues:=" , ["RF1","7"],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"DrivenTonesValues:=" , ["1Ghz"], "HarmonicsText:=" , "2F2 3F2 4F2 DC F2", ["NAME:SweepDefinition", "Variable:=" , "FNoi", "Data:=" , "LINC 1GHz 200GHz 10", "OffsetF1:=" , False, "Synchronize:=" , 0]])

Edit Oscillator Resonant Frequency Search [Nexxim]

Use: Edits an Oscillator Resonant Frequency Search setup

Syntax: EditOscResonantFrequencySearch(

STRING name,

ARRAY newsetup) // new setup data

Python Syntax	EditOscResonantFreqencySearch(<name>,<setupData[]>)
Python Example	oModule.EditOscResonantFrequencySearch("OscResonantFrequency", [

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "NAME:SimSetup",
    "DataBlockID:="         , 21,
    "SimSetupID:="          , 26,
    "OptionName:="          , "(Default Options)",
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "DC",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ],
            [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
            ]
        ],
        [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "NAME:NoiseOutputQuantities"
          ],
          "Name:="          , "OscResonantFrequency",
          "OscResonantFrequencyData:="      ,
              [1000,"1000000000","5000000000","1Ghz",True,1],
          "OscMaxkValues:=" , ["F1","7"
          "OscToneValues:=" , ["1Ghz"]
  ])
```

Edit PXF Setup[Nexxim]

Use: Edits a Periodic Transfer Function (PXF) setup

Syntax: EditPXFSetup(

```
STRING name,
ARRAY newsetup) // new setup data
```

Python Syntax	EditPXFSetup(<name>,<setupData[]>)

Python Example	oModule.EditPXFSetup("PXF",
	  [
	  "NAME:SimSetup",
	  "DataBlockID:=" , 21,
	  "SimSetupID:=" , 26,
	  "OptionName:=" , "(Default Options)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "3F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "4F1",
                "Selected:="            , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "DC",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "F1",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ]
    ],
    [
      "NAME:NoiseOutputQuantities"
    ],
    "Name:="          , "PXF",
    "PXFData:="       , [1,"HB",False,"HB","No",0],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "PXFMaxkValues:=" , ["F1","7"],
            "PXFTonesValues:="        , ["1Ghz"],
            "PXFRelHarmNums:="        , [],
            "PXFMaxSideband:="        , ["0"],
            "PXFSideband:="           , ["0"],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
            [
               "NAME:SweepDefinition",
               "Variable:="           , "F",
               "Data:="               , "LINC 10GHz 100GHz 10",
               "OffsetF1:="           , False,
               "Synchronize:="                , 0
            ],
            [
               "NAME:SweepDefinition",
               "Variable:="           , "F",
               "Data:="               , "LINC 10GHz 100GHz 10",
               "OffsetF1:="           , False,
               "Synchronize:="                , 0
            ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	])

Edit QuickEye Analysis

Modifies an existing QuickEye analysis.

Command: Double-click on the analysis in the project tree.

Syntax: Edit QuickEye Analysis (<string>, // name of analysis to edit

Array("NAME:SimSetupName",

"DataBlockID:=", <int>,

"SimSetupID:=", <int>,

"OptionName:=", <string>,

"AdditionalOptions:=", <string>,

"AlterBlockName:=", <string>,

"FilterText:=", <string>,

"AnalysisEnabled:=", <int>, // 1 if enabled, 0 if disabled

Array("NAME:OutputQuantities", <QuantityArray>, <QuantityArray>...)

Array("NAME:NoiseOutputQuantities", <QuantityArray>, <QuantityArray>...)),

"Name:=", <string>, // name for new analysis

"QuickEyeAnalysis:=", Array(<string>, // Input rise time

<string>, // Input low voltage

<string>, // Input high voltage

<string>, // bits per second

<string>, // number of FFE taps

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<string>, // Random Jitter Standard Deviation

<string>, // Delay

<string>, // Duty cycle distortion

<bool>, // true if unit interval, false if bits per second

<string>, // number of DFE taps

<bool>, // true to calculate FFE, false if weights are specified

<bool>, // true to calculate DFE, false if weights are specified

<string>, // DFE decision threshold

<string>, // DFE decision high

<string>, // DFE decision low

<bool>), // true if using specified equalization, false if disabled

Array("NAME:SweepDefinition",

"Variable:=", <string>,

"Data:=", <string>, // sweep

"OffsetF1:=", <bool>,

"Synchronize:=", <int>), // 1 to Synchronize, 0 otherwise

"FFEWts:=", Array(".5", "-2"), // optional, specified weights

"DFEWts:=", Array("2"))) // optional, specified weights

Return Value: <string> – // Name of the analysis after being modified

// If the name requested conflicts with the name of an existing

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

// analysis, the requested name is altered to be unique.

// The name returned reflects any change made to be unique.

Parameters: <QuantityArray>:

Array("NAME:Quantity",

"NodeType:=", <string>, // CompInst, Variable, Net, Harmonics, or Custom

"CompID:=", <string>,

"CompName:=", <string>,

"QuantityName:=", <string>,

"Selected:=", <bool>,

"UnitType:=", <string>,

"DataType:=", <string>, // Real, Complex, Integer, Enum, Char, Free, Array, Record

"CircuitInstanceID:=", <string)

Python Syntax	EditQuickEyeAnalysis(<data_block>)

Python Example	`newname = oModule.EditQuickEyeAnalysis (`
	`[`
	`"NAME:SimSetup",`
	`"DataBlockID:=",`
	`28,`
	`"SimSetupID:=", 3,`
	`"OptionName:=", "(Default Options)",`
	`"AdditionalOptions:=","",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AlterBlockName:=", "",
"FilterText:=", "",
"AnalysisEnabled:=",1,
   [
      "NAME:OutputQuantities",
      [
         "NAME:Quantity",
         "NodeType:=","CompInst",
         "CompID:=", "5",
         "CompName:=", "Level01_NPN_Model_5",
         "QuantityName:=", "I",
         "Selected:=", True,
         "UnitType:=", "NoUnit",
         "DataType:=","Real",
         "CircuitInstanceID:=", ""
      ],
      [
         "NAME:Quantity",
         "NodeType:=", "CompInst",
         "CompID:=", "10",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "CompName:=", "RES__10",
            "QuantityName:=", "I",
            "Selected:=", True,
            "UnitType:=", "NoUnit",
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ],
         [
            "NAME:Quantity",
            "NodeType:=", Net",
            "CompID:=", "",
            "CompName:=", "net_47",
            "QuantityName:=", "V",
            "Selected:=", True,
            "UnitType:=", "NoUnit",
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ]
      ],
      [
         "NAME:NoiseOutputQuantities"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
 ],
"Name:=", "MyQuickEyeAnalysis",
"QuickEyeAnalysis:=",
 [
    "5e-10", "0", "1", "1e-9", "2", "0", "1e-9", "0",
    True, "1", False, False,"0", "1", "-1", True
 ],
 [
    "NAME:SweepDefinition",
    "Variable:=", "Temp",
    "Data:=", "LINC 1cel 10cel 10",
    "OffsetF1:=", False,
    "Synchronize:=", 0),
    "FFEWts:=",
    [
       ".5", "-2"
    ]
    DFEWts:=",
    [
       "2"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]]]

Edit System FD Analysis [Nexxim]

Use: Edits a System Frequency Domain (FD) Analysis setup

Syntax: EditSystemFDAnalysis(

STRING name,

ARRAY newsetup) // new setup data

Python Syntax	EditSystemFDAnalysis(<name>,<setupData[]>)
Python Example	oModule.AddSystemFDAnalysis("SystemFDAnalysis", ["NAME:SimSetup", "DataBlockID:=" , 32, "SimSetupID:=" , 16, "OptionName:=" , "(Default Options)", "AdditionalOptions:=" , "", "AlterBlockName:=" , "", "FilterText:=" , "",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"AnalysisEnabled:=" , 1, ["NAME:OutputQuantities"], ["NAME:NoiseOutputQuantities"], "Name:=" , "SystemFDAnalysis", "SystemFDAnalysis:=" , [False]])

Edit Transient [Nexxim]

Use: Edits a Transient analysis setup

Syntax: EditTransient(

ARRAY setup) // modified setup data

Python Syntax	EditTransient(<modifiedsetupData[]>)
Python Example	oModule.EditTransient("NexximTransient1",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
[
  "NAME:SimSetup",
  "DataBlockID:="         , 10,
  "SimSetupID:="          , 11,
  "OptionName:="          , "(Default Options)",
  "AdditionalOptions:="   , "",
  "AlterBlockName:="      , "",
  "FilterText:="          , "",
  "AnalysisEnabled:="     , 1,
  [
          "NAME:OutputQuantities"
  ],
  [
          "NAME:NoiseOutputQuantities"
  ],
  "Name:="                , "NexximTransient1",
  "TransientData:="       , ["0.1ns","10ns"],
  "TransientNoiseData:="  , [False,"","",0,1,0,False,1],
  "TransientOtherData:="  , ["default"],
  [
          "NAME:SweepDefinition",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	"Variable:=" , "Temp", "Data:=" , "LINC 10cel 200cel 10", "OffsetF1:=" , False, "Synchronize:=" , 0]])

Edit TV Noise [Nexxim]

Use: Edits a TV Noise analysis setup

Syntax: EditTVNoise(

STRING setup

ARRAY setupdata) // modified setup data

Python Syntax	EditTVNoise(<setupName>,<setupData[]>)
Python Example	oModule.EditTVNoise(["NAME:SimSetup", "DataBlockID:=" , 20, "SimSetupID:=" , 21, "OptionName:=" , "(Default Options)",

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
    "AdditionalOptions:="   , "",
    "AlterBlockName:="      , "",
    "FilterText:="          , "",
    "AnalysisEnabled:="     , 1,
    [
            "NAME:OutputQuantities",
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "2F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ],
            [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "3F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
                ],
                [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "4F1",
                    "Selected:="            , True,
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
        [
                "NAME:Quantity",
                "NodeType:="            , "Harmonics",
                "CompID:="              , "",
                "CompName:="            , "",
                "QuantityName:="        , "DC",
                "Selected:="            , True,
                "UnitType:="            , "NoUnit",
                "NetlistFUT:="          , "",
                "DataType:="            , "Real",
                "TypeInfo:="            , ["",0,"",
                "IsUnconstrained:="     , False],
                "CircuitInstanceID:="   , ""
        ],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            [
                    "NAME:Quantity",
                    "NodeType:="            , "Harmonics",
                    "CompID:="              , "",
                    "CompName:="            , "",
                    "QuantityName:="        , "F1",
                    "Selected:="            , True,
                    "UnitType:="            , "NoUnit",
                    "NetlistFUT:="          , "",
                    "DataType:="            , "Real",
                    "TypeInfo:="            , ["",0,"",
                    "IsUnconstrained:="     , False],
                    "CircuitInstanceID:="   , ""
            ]
        ],
        [
          "NAME:NoiseOutputQuantities"
        ],
        "Name:="          , "TVNoise",
        "TVNoiseData:="           , [2,"","","","","HB",False,"HB","No",0],
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "TVNoiseMaxkValues:="     , ["F1","7","F2","7"],
            "TVNoiseTonesValues:="    , ["1Ghz","2Ghz"],
            "TVNoiseRelHarmNums:="    , [],
            "TVNoiseRefSideband:="    , ["1","1"],
            "HarmonicsText:=" , "2F1 3F1 4F1 DC F1",
            [
              "NAME:SweepDefinition",
              "Variable:="            , "F",
              "Data:="                , "LINC 10GHz 100GHz 10",
              "OffsetF1:="            , False,
              "Synchronize:="         , 0
            ],
            [
              "NAME:FSweep",
              "Variable:="            , "F",
              "Data:="                , "LINC 10GHz 100GHz 10",
              "OffsetF1:="            , False,
              "Synchronize:="         , 0
            ]
  ])
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Edit VerifEye Analysis

Edits an existing VerifEye analysis.

Command: Double-click on the analysis in the project tree.

Syntax: Edit VerifEye Analysis (<string>, // name of analysis to edit

Array("NAME:SimSetupName",

"DataBlockID:=", <int>,

"SimSetupID:=", <int>,

"OptionName:=", <string>,

"AdditionalOptions:=", <string>,

"AlterBlockName:=", <string>,

"FilterText:=", <string>,

"AnalysisEnabled:=", <int>, // 1 if enabled, 0 if disabled

Array("NAME:OutputQuantities", <QuantityArray>, <QuantityArray>...)

Array("NAME:NoiseOutputQuantities", <QuantityArray>, <QuantityArray>...)),

"Name:=", <string>, // name for modified analysis

"VerifEyeAnalysis:=", Array(<string>, // Input rise time

<string>, // Input low voltage

<string>, // Input high voltage

<string>, // bits per second

<string>, // number of FFE taps

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<string>, // Random Jitter Standard Deviation

<string>, // Delay

<string>, // Duty cycle distortion

<bool>, // true if unit interval, false if bits per second

<string>, // number of DFE taps

<bool>, // true to calculate FFE, false if weights are specified

<bool>, // true to calculate DFE, false if weights are specified

<string>, // DFE decision threshold

<string>, // DFE decision high

<string>, // DFE decision low

<bool>), // true if using specified equalization, false if disabled

Array("NAME:SweepDefinition",

"Variable:=", <string>,

"Data:=", <string>, // sweep

"OffsetF1:=", <bool>,

"Synchronize:=", <int>), // 1 to Synchronize, 0 otherwise

"FFEWts:=", Array(".5", "-2"), // optional, specified weights

"DFEWts:=", Array("2"))) // optional, specified weights

Return Value: <string> – // Name of the analysis after being modified

// If the name requested conflicts with the name of an existing

// analysis, the requested name is altered to be unique.

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

// The name returned reflects any change made to be unique.

Parameters: <QuantityArray>:

Array("NAME:Quantity",

"NodeType:=", <string>, // CompInst, Variable, Net, Harmonics, or Custom

"CompID:=", <string>,

"CompName:=", <string>,

"QuantityName:=", <string>,

"Selected:=", <bool>,

"UnitType:=", <string>,

"DataType:=", <string>, // Real, Complex, Integer, Enum, Char, Free, Array, Record

"CircuitInstanceID:=", <string)

Python Syntax	EditVerifEyeAnalysis(<data_block>)

Python Example	`newname = oModule.EditVerifEyeAnalysis (`
	`[`
	`"NAME:SimSetup",`
	`"DataBlockID:=",`
	`28,`
	`"SimSetupID:=", 3,`
	`"OptionName:=", "(Default Options)",`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
"AdditionalOptions:=","",
"AlterBlockName:=", "",
"FilterText:=", "",
"AnalysisEnabled:=",1,
   [
      "NAME:OutputQuantities",
      [
         "NAME:Quantity",
         "NodeType:=","CompInst",
         "CompID:=", "5",
         "CompName:=", "Level01_NPN_Model_5",
         "QuantityName:=", "I",
         "Selected:=", True,
         "UnitType:=", "NoUnit",
         "DataType:=","Real",
         "CircuitInstanceID:=", ""
      ],
      [
         "NAME:Quantity",
         "NodeType:=", "CompInst",
         "CompID:=", "10",
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
            "CompName:=", "RES__10",
            "QuantityName:=", "I",
            "Selected:=", True,
            "UnitType:=", "NoUnit",
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ],
         [
            "NAME:Quantity",
            "NodeType:=", Net",
            "CompID:=", "",
            "CompName:=", "net_47",
            "QuantityName:=", "V",
            "Selected:=", True,
            "UnitType:=", "NoUnit",
            "DataType:=", "Real",
            "CircuitInstanceID:=", ""
         ]
      ],
      [
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
        "NAME:NoiseOutputQuantities"
     ],
    "Name:=", "MyVerifEyeAnalysis",
    "VerifEyeAnalysis:=",
     [
        "5e-10", "0", "1", "1e-9", "2", "0", "1e-9", "0",
        True, "1", False, False,"0", "1", "-1", True
     ],
     [
        "NAME:SweepDefinition",
        "Variable:=", "Temp",
        "Data:=", "LINC 1cel 10cel 10",
        "OffsetF1:=", False,
        "Synchronize:=", 0),
        "FFEWts:=",
        [
           ".5", "-2"
        ]
        DFEWts:=",
        [
           "2"
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

	]]]

EditSweep [Nexxim]

Use: Edit a sweep

Syntax: EditSweep(

```
STRING setup) // setup name

ARRAY solvesetup_data) // setup data
```

GetAllSolutionSetups [Nexxim]

Use: Returns an array of the names of the simulation setups in the design

Syntax: GetAllSolutionSetups( )

Python Syntax	GetAllSolutionSetups( )
Python Example	`names = oModule.GetAllSolutionSetups()`

ListVariations [Nexxim]

Use: Get a list of solution variations for a given solution

Syntax: ListVariations(

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
STRING SolutionName) // solution name
```

Python Syntax	ListVariations(SolutionName)
Python Example	`variationArr = oModule.ListVariations("NWA1")`

RefreshMeshOverlays

Use: Refreshes mesh display for a setup

Syntax: RefreshMeshOverlays(

```
STRING setup) // setup name
```

RenameSweep [Nexxim]

Use: Renames a NexximSimulation setup

Syntax: RenameSweep(

```
STRING oldname) // old setup name
STRING newname) // new setup name
```

Example: `oModule. RenameSweep <oldname> <newname>`

Nexxim Linear Network Analysis

The following linear network analysis commands are available.

AddAnalysisOptions

AddSimulationSetup

EditAnalysisOptions

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditSimulationSetup

ExportForSpice

ExportForHSpice

RemoveAnalysisOptions

RemoveSimulationSetup

RenameAnalysisOptions

RenameSimulationSetup

Some of the following commands are available only in the "SolveSetups" module:

For example,

```
oDesign.GetModule("SolveSetups")

oModule.CommandName <args>
```

Add Analysis Options [Nexxim]

Use: Add an analysis option

UI Access	
Parameters	Name | Type | Description < Option_Data> | Array |
Return Value	None

Python Syntax	AddAnalysisOptions (option_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 Example	oModule.AddAnalysisOptions(["NAME:DataBlock", "DataBlockID:=" , 8, "Name:=" , "Nexxim Options", ["NAME:ModifiedOptions", "eye_sym_step_resp:=" , False, "tmi_flag:=" , "2.01"]])

AddSimulationSetup [Nexxim]

Use: Adds a new simulation setup

Command: None

Syntax: AddSimulationSetup <Simulationsetup_data>

Return Value: None

Parameters: <Simulationsetup_Data>

Edit Analysis Options [Nexxim]

Use: Edit an analysis option

Command: None

Syntax: EditAnalysisOptions

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
STRING name,
ARRAY <Option_Data>
```

Return Value: None

Python Syntax	EditAnalysisOptions(<name>,<Option_Data>[])

Python Example	`oModule.EditAnalysisOptions("Nexxim Options",`
	`[`
	`  "NAME:DataBlock",`
	`  "DataBlockID:="  , 8,`
	`  "Name:="  , "Nexxim Options",`
	`  [`
	`  "NAME:ModifiedOptions",`
	`  "eye_sym_step_resp:="  , False,`
	`  "tmi_flag:="  , "2.01",`
	`  "gen_timedomsmodel:="  , True`
	`  ]`
	`])`

EditSimulationSetup [Nexxim]

Use: Edit a simulation setup

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: None

Syntax: EditSimulationSetup <Simulationsetup_data>

Return Value: None

Parameters: <Simulationsetup_Data>

ExportForSpice [Nexxim]

Export matrix solution data to a file in the given spice format.

Command: None

Syntax: ExportForSpice <theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fittingError> <maximumOrder> <useCommonGround> <doPassivityCheck>

Return Value: None

Parameters: <theVariationKey>

Type: string

<sourceNames>

Type: array

<theType: 0=PSpice, 2=MaxwellSpice, 3=Spectre>

Type: long

<bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: long

<fullwaveSpiceFilename>

Type: string

<lumpedElementFilename>

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

<Unused>

Type: string

<Unused>

Type: string

<partialFractionFilename>

Type: string

<fittingError [Default:0.5]>

Type: double

<maximumOrder [Default:200]>

Type: int

<useCommonGround [Default:FALSE]>

Type: boolean

<doPassivityCheck [Default:FALSE]>

Type: boolean

Python Syntax	ExportForSpice(<theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fittingError> <maximumOrder> <useCommonGround> <doPassivityCheck>)
Python Example	

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

```
oDesign.ExportForSpice( "",
[
    "LNA:LNA"
] ,
0,
0,
"C:/Documents and Settings/Marcia.BMT/MyDocuments/Code Modifications/Test" & " Designs/s-params_fws_fws.lib",
"C:/Documents and Settings/Marcia.BMT/MyDocuments/Code Modifications/Test" & " Designs/s-params_fws_
lfws.lib",
"",
"",
"",
"",
0.52,
210,
1,
1)
```

ExportForHSpice [Nexxim]

Export matrix solution data to an HSpice subcircuit.

Command: None

Syntax: ExportForHSpice <theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fitting Error> <maximumOrder> <useCommonGround> <doPassivityCheck>

Return Value: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Parameters: <theVariationKey>

Type: string

<sourceNames>

Type: array

<theType: 1=HSPICE>

Type:long

<bandwidth: 0=Low Bandwidth, 1=Broad Bandwidth>

Type: long

<fullwaveSpiceFilename>

Type: string

<lumpedElementFilename>

Type: string

<Unused>

Type: string

<Unused>

Type: string

<partialFractionFilename>

Type: string

<fittingError [Default:0.5]>

Type: double

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

<maximumOrder [Default: 200]>

Type: int

<useCommonGround [Default: FALSE]>

Type: boolean

<doPassivityCheck [Default: FALSE]>

Type: boolean

Python Syntax	ExportForHSpice(<theVariationKey> <sourceNames> <theType> <bandwidth> <fullwaveSpiceFilename> <lumpedElementFilename> <Unused> <Unused> <partialFractionFilename> <fitting Error> <maximumOrder> <useCommonGround> <doPassivityCheck>)
Python Example	`oDesign.ExportForHSpice( "", ["Setup1:Sweep 1"], 1, 0, "C:/Projects/MyTRL_fws.sp", "", "", "", "", 0.5, 0, 200, 0, 0)`

RemoveAnalysisOptions [Nexxim]

Use: Remove an analysis option

Command: None

Syntax: RemoveAnalysisOptions <Option_Data>

Return Value: None

Parameters: `<Option_Data>`

RemoveSimulationSetup [Nexxim]

Use: Remove a simulation setup

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: RemoveSimSetup <Simulationsetup_data>

Return Value: None

Parameters: <Simulationsetup_Data>

Python Syntax	RemoveSimSetup(<setupName1>,<setupName2>...)
Python Example	`oDesign.RemoveSimSetup("QuickEyeAnalysis", "VerifyEyeAnalysis")`

RenameAnalysisOptions[Nexxim]

Use: Rename an analysis option

Command: None

Syntax: RenameAnalysisOptions <oldname> <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Example: oModule.RenameAnalysisOptions <oldname> <newname>

RenameSimulationSetup [Nexxim]

Use: Rename a simulation setup

Command: None

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Syntax: RenameSimulationSetup <oldname> <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Example: oModule.RenameSimulationSetup "oldname", "newname"

Interface Ports And Sources Commands

The following Ports and Sources commands for Nexxim are available.

ChangePortProperty

ChangeSourceProperty

DeletePort

DeleteSource

GetAllPorts

GetAllSources

RenameSource

ChangePortProperty [Nexxim]

Change a port property

UI Access	Right click port > Properties		
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.

	<Port Name>	string	port name
	<Port Info>	array	Array of attributes
	<Port Properties>	array	Array of attributes and values
Return Value	None		

Python Syntax

```
ChangePortProperty(<portName> [<portInfo>] [<portProperties>])
```

Python Example

```
oDesign.ChangePortProperty(

"PortName", [ "NAME:PortName", "IIPortName:=", "PortName", "SymbolType:=", 0],[ "NAME:Properties", ["NAME:NewProps", ["NAME:term", "PropType:=", "TextProp", "OverridingDef:=", True, "Value:=", "Model1"]], ["NAME:ChangedProps", ["NAME:TerminationData", "Value:=", "Zo"], ["NAME:pnum", "Value:=", "2"], ["NAME:noisetemp", "Value:=", "16.85"] ] ] )
```


Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ChangeSourceProperty [Nexxim]

Change a source property.

UI Access	Right click source > Properties
Parameters	<table><tr><th>Name</th><th>Type</th><th>Description</th></tr><tr><td><Source_Name></td><td>string</td><td>Source name</td></tr><tr><td><Properties></td><td>array</td><td>Array of source attributes and their values</td></tr></table>
Return Value	None

Python Syntax	ChangeSourceProperty(<sourceName>,[<Properties])

Python Example	<pre>oDesign.ChangeSourceProperty("SourceName",
	 [["NAME:Properties",
	  ["NAME:NewProps",
	   ["NAME:noise", "PropType:=", "TextProp", "OverridingDef:=", True, "UserDef:=",
	"Value:=", "SourceNoise1"],
	   ["NAME:mod", "PropType:=", "TextProp", "OverridingDef:=", True, "UserDef:=", T
	"Value:=", "SourceModulation1"]]],
	  ["NAME:DataBlocks",
	   ["NAME:NewDataBlk",
	    ["NAME:DataBlock", "Name:=", "SourceNoise1", "noisedata:=", ["FDEV=1K
	"RFLO=2 ", "NCOR=2 2 "]],</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:DataBlock", "Name:=", "SourceModulation1" "ModulationSourceData "Butterworth", "Br=1.2288MHz", "Dly=0.5", "Iasc=1", "Qasc=1"]]]]])

DeletePort [Nexxim]

Use: Delete a port

Command: None

Syntax: DeletePort <Port_Name>

Return Value: None

Parameters: <Port_Name>

Type: string

Python Syntax	DeletePort(<portName>)
Python Example	oDesign.DeletePort("Port1")

DeleteSource [Nexxim]

Use: Delete the source

Command: None

Syntax: DeleteSource <srcname>

Return Value: None

Parameters: <srcname>

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.

Python Syntax	DeleteSource(<sourceName>)
Python Example	`oDesign.DeleteSource("VoltageSource1")`

GetAllPorts [Nexxim]

Use: Returns a list of all port names in the Nexxim Design

Syntax: GetAllPorts

Return Value: Array of Strings

Parameters: None

Python Syntax	GetAllPorts()
Python Example	`portsArr = oDesign.GetAllPorts()`

GetAllSources [Nexxim]

Use: Returns a list of all sources names in the Nexxim Design

Syntax: GetAllSources

Return Value: Array of Strings

Python Syntax	GetAllSources()
Python Example	`sourceArr = oDesign.GetAllSources()`

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RenameSource [Nexxim]

Use: Rename the source

Command: None

Syntax: RenameSource <oldname>, <newname>

Return Value: None

Parameters: <oldname>

Type: string

<newname>

Type: string

Python Syntax	RenameSource(<oldName>,<newName>)
Python Example	`oDesign.RenameSource("VoltageSource1", "MySource1")`

Nexxim Component Manager Commands

The following Component Manager commands for Nexxim are available.

ImportSandWComponent

ImportXparamComponent

ImportSandWComponent

Use: Import S-element or W-element

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

Command: None

Syntax: ImportSandWComponent <Files><Options>

Return Value: Name of the component that is created

Parameters: `<Files>, <Options>`

Type: array

ImportXparamComponent

Use: Import X-Parameter component

Command: None

Syntax: ImportXparamComponent <File>

Return Value: Name of the component that is created

Parameters: `<File>`

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.

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

ChangeProperty 6-9, 7-12

ExportModelImagetoFile 9-54

GetProperties 2-72, 6-39, 7-54

GetPropertyValue 2-73, 6-40, 7-55, 9-75

GetPropEvaluatedValue 6-36, 7-51, 11-40, 11-114

GetPropSIValue 6-38, 7-53, 11-42, 11-115

3D Modeler editor object commands

GetObjPath 11-39, 11-113

A

Add 17-36

AddAllEyeMeasurements 9-4

AddAlterBlockNexxim 17-37

AddAMIAnalysisNexxim 17-38

AddAnalysisOptions 17-170

AddCartesianLimitLine 9-4

AddCartesianLimitLineFromCurve 9-5

AddCartesianLimitLineFromEquation 9-7

AddCartesianXMarker 9-8

AddCartesianYMarker 9-9

AddCartesianYMarkerToStack 9-10

AddDataset 6-4, 7-5

AddDCAnalysisNexxim 17-39

AddDeltaMarker 9-10

AddDeviceNoiseDataBlock 17-26

AddDiffPair 14-3

AddDynamicLink 7-6

AddEnvelopeSetupNexxim 17-41

AddHB1Tone 17-46

AddHBnTone 17-51

AddHSPICETransientNexxim 17-55

AddLibRefDataBlock 17-24

AddLinearNetworkAnalysisNexxim 17-57

AddMarker 9-11

AddMaterial 6-5

AddMaxwellLink 7-8, 7-9

AddMenuProp 2-48

AddMenuProp2 2-49

AddMessage 4-6

AddModelingProperties 7-8

AddNetlistDataBlock 17-24

AddNote 9-12

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddOscillatorAnalysisNexxim 17-59

AddOscillatorNtoneAnalysisNexxim 17-64

AddOscillatorResonantFrequencySearchNexxim 17-69

AddPinGrounds 16-36

AddPinIPorts 16-38

AddPinPageConnectors 16-39

AddPrintToAuditDataBlock 17-25

AddProp 2-51

AddProp2 2-54

AddPXFSetupNexxim 17-74

AddQuickEyeAnalysis 17-79

AddSimulationSetup 17-171

AddStateVariableDataBlock 17-25

AddSubstrateDataBlock 17-26

AddSweep 17-100

AddSystemFDAnalysisNexxim 17-86

AddTemperatureDataBlock 17-23

AddTraceCharacteristics 9-14

AddTraces 9-15

AddTransientNexxim 17-87

AddTVNoiseNexxim 17-88

AddVerifyEyeAnalysis 17-94

AlignHorizontal 16-42

AlignVertical 16-43

Analysis module commands

AnalyzeAll 7-10

AnalyzeAllNominal 7-10

CopyDrivenSetup 10-2

CopyEigenSetup 10-3

CopySetup 10-2, 11-7, 11-87

CopySweep 10-3

DeleteDrivenSweep 10-4

DeleteSetups 10-5

DoesSweepSetupExists 10-5

EditDrivenSweep 10-6

EditFrequencySweep 10-8

EditSetup 10-9, 11-9, 11-89

GetSetupCount 10-24

GetSetups 10-25

GetSweepCount 10-26

GetSweeps 10-27

InsertDrivenSweep 10-28

InsertFrequencySweep 10-32

InsertSetup 10-36, 11-46, 11-119

InsertSetup Transient 10-57

PasteDrivenSetup 10-60

PasteEigenSetup 10-61

PasteSetup 10-61

PasteSweep 10-62

RenameDrivenSweep 10-62

RenameSetup 10-63

ResetToTimeZero 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.

RevertAllToInitial 10-64

RevertAllToZeroDisplacement 10-64

RevertSetupToInitial 10-65

RevertSetupToZeroDisplacement 10-65

SolveSetup 10-66

Analysis Setup Module Script Commands 10-1

Analyze 17-2, 17-2, 17-101, 17-101

AnalyzeAllNominalNexxim 17-100

AnalyzeSweep 17-101

Ansoft Application Object commands 3-1

GetAppDesktop 3-2

ApplyReportTemplate 9-17

C

Cascade 14-4

ChangePortProperty 7-11, 17-179

ChangeProperty 2-41, 6-9, 7-12, 9-17, 16-83

ChangeSourceProperty 7-16, 17-181

ClearAllMarkers 9-24

ClearAllTraceCharacteristics 9-24

ClearDiffPairs 14-5

ClearMessages 4-7, 6-13

Clone 14-6

CloneMaterial 6-13

CloneReportsFromDatasetSolution 9-25

Close 6-14, 14-7

CloseAllWindows 4-10

CloseEditor 16-45

CloseProject 4-11

CloseProjectNoForce 4-12

Combine 14-8

CompInstance Functions 54

CompInstance Script Commands (multiple)

conventions

scripting help 1-1

Copy 16-46

CopyData 16-47

CopyDesign 6-15

CopyDrivenSetup 10-2

CopyEigenSetup 10-3

CopyEyeItemAsCommand 7-17, 17-3

CopyItemCommand 7-18

CopyPlotSettings 9-26

CopyReportDefinition 9-26

CopyReportsData 9-27

Copyright and Trademark Information 2

CopySetup

Analysis module command 10-2, 11-7, 11-87

Optimetrics module command 10-2, 11-7, 11-87

CopySubdesign 16-47

CopySweep 10-3

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

CopyTraceDefinitions 9-27
CopyTracesData 9-28
Count 4-12
CPython 1-44
CreateNPort 16-20
CreateOutputVariable 8-1
CreatePage 16-48
CreateReport 7-18, 9-29
CreateReportFromTemplate 9-41
CreateReportofAllQuantities 9-42
CreateUserDefinedSolution 13-1
CutDesign 6-15

D

dataset commands
AddDataset 6-4, 7-5
DeleteDataset 6-16, 7-31
EditDataset 6-18, 7-35
ExportDataset 6-26, 7-39
ImportDataset 6-44, 7-64
DeactivateOpen 16-50
DeactivateShort 16-50
Deembed 14-9
DeembedBack 14-10
DeembedFront 14-11
Delete 16-51, 17-101
DeleteAllReports 9-43
DeleteDataset 6-16, 7-31
DeleteDesign 6-17
DeleteDesignInstance 7-32, 7-32, 17-3, 17-3
DeleteDrivenSweep 10-4
DeleteMarker 9-43
DeleteOutputVariable 7-32, 8-3
DeletePort 7-33, 17-182
DeleteProject 4-13
DeleteReport 9-44
DeleteSetups 10-5
Analysis module command 10-4
Optimetrics module command 11-8, 11-87
DeleteSolutionVariation 7-33
DeleteSource 7-34, 17-182
DeleteSweep 17-102
DeleteTraceCharacteristics 9-44
Deletetraces 9-45
DeleteUserDefinedSolutions 13-2
Design object commands 9-65, 9-65, 9-67, 9-67, 9-68, 9-80
AddCartesianLimitLine 9-4
AddCartesianXMarker 9-8
AddCartesianYMarker 9-9
AddDeltaMarker 9-10
AddDynamicLink 7-6, 7-8, 7-9
AddMarker 9-11
AddNote 9-12
AddQuickEyeAnalysis 17-79
AddTraceCharacteristics 9-14
AddTraces 9-15

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

AddVerifyEyeAnalysis 17-94

ClearAllMarkers 9-24

ClearAllTraceCharacteristics 9-24

CloseAllWindows 4-10

CopyEyeItemAsCommand 7-17, 17-3

CopyReportDefinition 9-26

CreateOutputVariable 8-1

CreateReportFromTemplate 9-41

DeleteAllReports 9-43

DeleteOutputVariable 7-32, 8-3

DeleteReport 9-44

DeleteTraces 9-45

DoesOutputVariableExist 8-3

EditMarker 9-49

EditOutputVariable 7-37, 8-4

EditQuickEyeAnalysis 17-147

EditVerifEyeAnalysis 17-162

ExportModelMeshToFile 9-57

ExportPlot3DToFile 9-58

ExportReport 7-41, 9-59

ExportToFile 9-62

GetDisplayType 9-71

GetLibraryDirectory 4-24

GetModule 7-46

GetName 7-47, 9-74, 11-38, 11-112

GetObjPath 7-49, 9-75

GetOutputVariableValue 7-49, 8-7

GetProjectDirectory 4-28

GetProperties 2-72, 6-39, 7-54

GetPropertyValue 2-73, 6-40, 7-55, 9-75

GetPropEvaluatedValue 6-36, 7-51, 11-40, 11-114

GetPropNames 7-52

GetPropSIValue 6-38, 7-53, 11-42, 11-115

GetRegistryInt 4-29, 4-63

GetRegistryString 4-30, 4-64

GetTempDirectory 4-33

GetVariables 2-75, 6-43, 7-59

GetVariableValue 2-75, 6-42, 7-58

GetVariationVariableValue 7-59

GetVersion 4-34

ImportIntoReport 9-94

IsFeaturEnabled 4-35

PasteDesign 7-67

PasteReports 9-98

PasteReportsWithLegacyNames 9-99

PasteTracesWithLegacyNames 9-100

Redo 7-69

RenameDesignInstance 7-70

RenameReport 9-101

RenameTraces 9-101

RunToolkit 6-52, 7-72

SavePlotSettingsAsDefault 9-103

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetActiveEditor 7-72
SetLibraryDirectory 4-54
SetProjectDirectoryVBCommand> 4-54
SetPropertyValue 2-76, 6-59, 7-74
SetTempDirectory 4-57
SetVariableValue 2-77, 6-60, 7-75
SimulateLink 7-76
Solve 7-76
UpdateAllReports 9-105
Design Object Script Commands 7-1
Desktop Commands For Registry Values 4-61
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
GetActiveProject 4-17
GetAutosaveEnabled 4-18
GetBuildTimeDateString 4-18
GetChildNames 6-29, 7-42
GetChildTypes 6-30, 7-44
GetChildTypes Optimetrics 11-38, 11-112
GetCustomMenuSet 4-19
GetDesigns 6-32
GetDesktopConfiguration 4-22
GetDistributedAnalysisMachines 4-22
GetDistributedAnalysisMachinesForDesignType 4-23
GetProjectList 4-28
GetProjects 4-26, 4-29
GetPropNames 6-37, 11-41, 11-115
GetSchematicEnvironment 4-31
KeepDesktopResponsive 4-36
LaunchJobMonitor 4-37
NewProject 4-37
OpenProject 4-39
PageSetup 4-40
PauseRecording 4-40
PauseScript 4-41
Print 4-41
QuitApplication 4-42
RefreshJobMonitor 4-42
ResetLogging 4-43
RestoreWindow 4-45
ResumeRecording 4-45
RunProgram 4-46
RunScript 4-47
SelectScheduler 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.

SetActiveProject 4-51
SetActiveProjectByPath 4-51
SetCustomMenuSet 4-52
SetDesktopConfiguration 4-53
SetSchematicEnvironment 4-57
Sleep 4-59
SubmitJob 4-59
TileWindows 4-60
Desktop Object Script Commands 4-1
Desktop Scripting Conventions 1-57
DisableDiffPairs 14-12
DisplayBiasPointInfo 17-102
DistributedAnalyzeSetup, Optimetrics module command 11-8, 11-88
DoesOutputVariableExist 8-3
DoesSupportTraceCharacteristics 9-46
DoesSweepSetupExists 10-5
DumpAllReportsData 9-47
DynamicMeshOverlays 17-102

E

Edit 17-103
EditAlterBlockNexxim 17-103
EditAMIAnalysisNexxim 17-104
EditAnalysisOptions 17-171
EditCartesianXMarker 9-47
EditCartesianYMarker 9-48
EditCircuitSettings 10-6
EditDataset 6-18, 7-35
EditDCAnalysisNexxim 17-106
EditDrivenSweep 10-6
EditEnvleopeSetupNexxim 17-107
EditFrequencySweep 10-8
EditHB1Tone 17-113
EditHBnTone 17-118
EditHSPICETransientNexxim 17-123
EditImportData 7-36, 17-4
EditLibRefDataBlock 17-27
EditLinearNetworkAnalysisNexxim 17-124
EditMarker 9-49
EditMaterial 6-19
EditNetlistDataBlock 17-28
EditNotes 7-37, 17-5
Editor object commands
SetPropertyValue 2-76, 6-59, 7-74
Editor Scripting IDs 16-4
EditOscillatorAnalysisNexxim 17-126
EditOscillatorNtoneAnalysisNexxim 17-131
EditOscillatorResonantFrequencySearchNexxim 17-136
EditOutputVariable 7-37, 8-4
EditPrintToAuditDataBlock 17-29

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

EditPXFSetupNexxim 17-141

EditQuickEyeAnalysis 17-147

EditSetup 10-9, 11-9, 11-89

optimization command 11-21, 11-150

parametric command 11-20, 11-144

sensitivity command 11-25, 11-159

statistical command 11-28, 11-166

EditSimulationSetup 17-172

EditSweep 17-168

EditSystemFDAnalysisNexxim 17-153

EditTemperatureDataBlock 17-27

EditTransientNexxim 17-154

EditTVNoiseNexxim 17-156

EditUserDefinedSolution 13-3

EditVerifEyeAnalysis 17-162

ElectricRuleCheck 16-53

EnableAutoSave 4-16

EnableDiffPairs 14-13

EnableSetup 11-29, 11-100

ExecuteScript 2-56

ExportCircuit 10-20

ExportCitiFile 14-13

ExportDataset 6-26, 7-39

ExportDOELocalSensitivity 11-109

ExportDOELocalSensitivityCurve 11-110

ExportDOEResponseCurve 11-107

ExportDOEResponseCurveSlices 11-108

ExportDOEResponseSurface 11-108

ExportDXConfigFile, Optimetrics module command 11-30, 11-101

ExportEyeMaskViolation 9-49

ExportForSpice 7-39

Linear Network Analysis command 17-5, 17-8, 17-173, 17-175

Nexxim Netlist command 17-5, 17-8, 17-173, 17-175

ExportFullWaveSpice 14-15

ExportImageToFile 9-50

ExportMaterial 6-27

ExportMatlab 14-19

ExportModelMeshToFile 9-57

ExportNetlist 7-40, 17-10

ExportOptimetricsProfile, Optimetrics module command 11-30, 11-101

ExportOptimetricsResults, Optimetrics module command 11-31, 11-102

ExportOptionsFiles 4-17

ExportOutputVariables 8-5

ExportParametricResults, Optimetrics module command 11-32, 11-103

ExportPlot3DToFile 9-58

ExportReport 7-41, 9-59

ExportReportDataToFile 9-60

ExportRespSurfaceMinMaxTable 11-33, 11-104

ExportRespSurfaceRefinePoints 11-34, 11-105

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportRespSurfaceResponsePoints 11-35, 11-106

ExportRespSurfaceVerificationPoints 11-35, 11-106

ExportSpreadsheet 14-22

ExportTableToFile 9-60

ExportToFile 9-61

ExportTouchstone 14-24

ExportTouchstone2 14-27

ExportUniformPointsToFile 9-63

Extract 14-29

F

FindElements 16-55

FitToBorder 16-57

FlipHorizontal 16-58

FlipVertical 16-59

G

general method script commands 16-33

GenerateVariationData Parametric, parametric command 11-36, 11-146

GetActiveDesign 6-27

GetActiveProject 4-17

GetAllCategories 9-65

GetAllPorts 17-183

GetAllQuantities 9-66

GetAllReportNames 9-66

GetAllSolutionSetups 17-168

GetAllSources 17-183

GetApplication 2-56

GetArrayVariables 2-72, 6-28

GetAutoSaveEnabled 4-18

GetAvailableDisplayTypes 9-67

GetAvailableReportTypes 9-67

GetAvailableSolutions 9-68

GetBuildDateTimeString 4-18

GetCallback 2-57

GetChangedProperty 2-58

GetChild Object (Report Setup), Report module command 9-69

GetChildNames 6-29, 7-42

GetChildNames (Optimetrics), Report module command 9-68

GetChildObject Design 6-29, 7-43

GetChildTypes 6-30, 7-44

GetChildTypes Optimetrics 11-38, 11-112

GetChildTypes ReportSetup 9-70

GetComponentName 54

GetComponentPinLocation 16-93

GetCurvePropServerName 9-70

GetDataUnits 9-82

GetDefinitionManager 6-31

GetDependentFiles 6-31

GetDescription 2-58

GetDesign 2-59, 2-59

GetDesignID 7-44

GetDesigns 6-32

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetDesignType 7-45

GetDesignVariableNames 9-83

GetDesignVariableUnits 9-84

GetDesignVariableValue 9-85

GetDesignVariationKey 9-86

GetDisplayType 9-71

GetDistributedAnalysisMachines 4-22

GetDistributedAnalysisMachinesForDesignType 4-23

GetDocumentNames 12-5

GetDynLinkIntrinsicVariables 9-72

GetDynLinkQtyValueState 9-72

GetDynLinkTraces 9-73

GetDynLinkVariableValues 9-73

GetEditor 2-59, 7-45

GetEditor [Nexxim] 17-12

GetEditorName [Schematic] 16-94

GetEvaluatedPropertyValue 16-84

GetEvaluatedText 2-60

GetExeDir 4-23

GetFileName 2-60

GetFrequencies 14-30

GetFrequencyCount 14-30

GetHidden 2-61

GetImagDataValues 9-86

GetInstanceID 55

GetInstanceName 55

GetLibraryDirectory 4-24

GetModule 7-46, 17-10

GetName 6-34, 7-47, 9-74, 11-38, 11-112, 14-31, 17-11

GetNumPages 16-95

GetObjPath 6-35, 7-49, 9-75, 11-39, 11-113

GetOutputVariableValue 7-49, 8-7

GetParentDesign 56

GetPath 6-35

GetPerQuantityPrimarySweepValues 9-87

GetPersonalLibDirectory 4-27

GetPortCount 14-32

GetPortNumber 14-33

GetPostProcSettings 14-34

GetProcessID 4-27

GetProgress 2-62, 2-62

GetProjectDirectory 4-28

GetProjectList 4-28

GetProjects 4-26, 4-29

GetProperties 2-72, 6-39, 7-54, 16-84

GetPropertyValue 2-73, 6-40, 7-55, 9-75

GetPropEvaluatedValue 6-36, 7-51, 11-40, 11-114

GetPropHost 2-62, 2-62, 7-52, 7-52, 56

GetPropNames 6-37, 7-52, 9-77, 11-41, 11-115

GetPropServerName 57

GetPropServers 2-62

GetPropSIValue 6-38, 7-53, 11-42, 11-115

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

GetPropTabType 2-63

GetPropValue Optimetrics 11-43, 11-116

GetPropValue Project 6-39, 7-54

GetPropValue Reporter 9-77

GetQtyExpressionsForSourceTrace 9-78

GetReadOnly 2-63

GetRealDataValues 9-88

GetRegistryInt 4-29, 4-63

GetRegistryString 4-30, 4-64

GetReportSummaryForRegressionTesting 9-79

GetReportTraceNames 9-78

GetResultsDirectory 7-57, 17-12

GetRunStatus 2-64

GetSetupCount 10-24

GetSetupCount, Analysis module command 10-24

GetSetupNames 10-26

GetSetupNames (Optimetrics), Optimetrics module command 11-36, 11-37, 11-43, 11-110, 11-111, 11-117

GetSetupNamesByType (Optimetrics), Optimetrics module command 11-44, 11-118

GetSetups 10-25, 10-25

Analysis module command 10-25

GetSolutionContexts 9-79

GetSolutionDataPerVariation 9-80

GetSolutionVariation 14-34

GetSourceData 7-58

GetSweepCount 10-26

GetSweepCount, Analysis module command 10-26

GetSweepNames 9-89

GetSweeps 10-27, 10-27

Analysis module command 10-27

GetSweepUnits 9-90

GetSweepValues 9-90

GetSysLibDirectory 4-33

GetTabTypeName 2-64

GetTempDirectory 4-33

GetTemperatureDataBlock 17-31

GetText 2-65

GetTool 4-67

GetTopDesignList 6-42

GetUserDefinedSolutionNames 13-4

GetUserDefinedSolutionProperties 13-4

GetUserLibDirectory 4-34

GetValue 2-66

GetVariables 2-75, 6-43, 7-59

GetVariableValue 2-75, 6-42, 7-58

GetVariation 14-35

GetVariationVariableValue 7-59

GetVersion 4-34

GridSetup 16-60

GroupPlotCurvesByGroupingStrategy 9-93

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

H

HasSameData 14-36

hierarchy of variables in HFSS 1-50

I

Icepak boundary condition commands

ExportDOELocalSensitivity 11-109

ExportDOELocalSensitivityCurve 11-110

ExportDOEResponseCurve 11-107

ExportDOEResponseCurveSlices 11-108

ExportDOEResponseSurface 11-108

ImportIDX 4-77

ImportANF 4-68, 4-68

ImportAutoCAD 4-69

ImportAWRMicrowaveOffice 4-70

ImportData 7-60, 17-13

ImportDataFilePath 7-64, 17-16

ImportDataset 6-44, 7-64

ImportEDB 4-71

ImportExport Tool 4-67

ImportExtracta 4-71

ImportGDSII 4-72, 4-73

ImportIDF 4-74, 4-77

ImportIDX 4-77

ImportIntoReport 9-94

ImportIPC 4-80

ImportODB 4-80

ImportOutputVariables 8-8

ImportReportDataIntoReport 9-95

ImportSandWComponent 17-185

ImportSetup, Optimetrics module command 11-45, 11-118

ImportXFL 4-81

ImportXparamComponent 17-185

Information Method List 16-88

InsertDesign 6-45, 7-66, 17-17

InsertDesignWithWorkflow 6-46

InsertDrivenSweep 10-28

InsertFrequencySweep 10-32

InsertSetup 10-36, 11-46, 11-119

Analysis module command 10-36, 11-46, 11-119

optimization command 11-71, 11-154

parametric command 11-67, 11-146

sensitivity command 11-77, 11-163

statistical command 11-79, 11-167

InsertSetup Transient, Analysis module command 10-57

interface

ports and sources commands 17-179

IronPython 1-2

IsDataComplex 9-91

IsFeatureEnabled 4-35

IsPerQuantityPrimarySweep 9-92

IsValueConstant 2-66

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

K

KeepDesktopResponsive 4-36

L

LaunchJobMonitor 4-36

Layout Scripting 1-59

ListVariations 17-168

LoadSolution 14-38

M

material commands

AddMaterial 6-5

CloneMaterial 6-13

EditMaterial 6-19

ExportMaterial 6-27

RemoveMaterial 6-49

method format, create schematic objects 16-1

module script commands

general method 16-33

modules in HFSS scripting 1-50

MovePlotCurvestoGroup 9-95

MovePlotCurvestoNewGroup 9-96

N

Named Arguments 1-57

NameNets 16-63

NdExplorer

Script Commands 14-1

NewProject 4-37

Nexxim Component Manager Commands 17-184

Nexxim Data Block Commands 17-22

Nexxim Linear Network Analysis 17-169

Nexxim Netlist Scripting 17-1

Nexxim Scripting 17-1

Nexxim Simulation Setup Commands 17-34

O

oAnsoftApp object 1-50

object-oriented scripting 2-1

oDesign object 1-50

oDesktop object 1-50

oEditor object 1-50

oModule object 1-50

Open 14-38

OpenAndConvertProject 4-38

OpenProject 4-39

OpenWindowForAllReports 9-97

OpenWindowForReports 9-97

oProject object 1-50

Optimetrics module commands

DeleteSetups 11-8, 11-88

DistributeAnalyzeSetup 11-8, 11-88

EnableSetup 11-29, 11-100

ExportDXConfigFile 11-30, 11-101

ExportOptimetricsProfile 11-30, 11-101

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

ExportOptimetricsResults 11-31, 11-102

ExportParametricResults 11-32, 11-103

ExportRespSurfaceMinMaxTable 11-33, 11-104

ExportRespSurfaceRefinePoints 11-34, 11-105

ExportRespSurfaceResponsePoints 11-35, 11-106

ExportRespSurfaceVerificationPoints 11-35, 11-106

GetChildNames 11-36, 11-37, 11-110, 11-111

GetOptimetricResult 11-40, 11-113

GetSetupNames 11-43, 11-117

GetSetupNamesByType 11-44, 11-118

ImportSetup 11-45, 11-118

RenameSetup 11-82, 11-141

SolveSetup 10-2, 11-7, 11-84, 11-84, 11-87, 11-143, 11-143

Optimetrics Module Script Commands 11-1

optimization commands

EditSetup 11-21, 11-150

InsertSetup 11-71, 11-154

Optimization Script Commands 11-150

output variable commands

CreateOutputVariable 8-1

DeleteOutputVariable 7-32, 8-3

DoesOutputVariableExist 8-3

EditOutputVariable 7-37, 8-4

GetOutputVariableValue 7-49, 8-7

Output variable commands

ExportOutputVariables 8-5

ImportOutputVariables 8-8

Output Variable Script Commands 8-1

P

PageBorders 16-64

PageSetup (Layout Editor) 4-40

Pan 16-65

parametric commands

EditSetup 11-20, 11-144

GenerateVariationData Parametric 11-36, 11-146

InsertSetup 11-67, 11-146

Parametric Script Commands 11-144

Paste 6-48, 16-65

Paste (Project Object) 6-48

PasteData 16-66

PasteDesign 7-67

PasteDesign (Schematic Editor) 16-67

PasteDrivenSetup 10-60

PasteEigenSetup 10-61

PasteItemCommand 7-68, 17-18

PastePlotSettings 9-98

PasteReports 9-98

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

PasteReportsWithLegacyNames 9-99

PasteSetup

Analysis module command 10-61

Optimetrics module command 11-82, 11-141

PasteSweep 10-62

PasteTraces 9-100

PasteTracesWithLegacyNames 9-100

PauseRecording 4-40

PauseScript 4-41

Print 4-41

Project object commands

AddDataset 6-4, 7-5

AddMaterial 6-5

CloneMaterial 6-13

Close 6-14

CopyDesign 6-15

CutDesign 6-15

DeleteDataset 6-16, 7-31

DeleteDesign 6-17

EditDataset 6-18, 6-26, 6-43, 7-35, 7-38, 7-64

EditMaterial 6-19

ExportMaterial 6-27

GetActiveDesign 6-27

GetArrayVariables 2-72, 6-28

GetChildObject Design 6-29, 7-43

GetDependentFiles 6-31

GetDesign 2-59

GetDesignID 7-44, 7-46

GetDesignType 7-45

GetName 6-34

GetObjPath 6-35

GetPath 6-35

GetProperties 2-72, 6-39, 7-54

GetPropertyValue 2-73, 6-40, 7-55, 9-75

GetPropEvaluatedValue 6-36, 7-51, 11-40, 11-114

GetPropSIValue 6-38, 7-53, 11-42, 11-115

GetPropvalue Optimetrics 11-43, 11-116

GetPropvalue Project 6-39, 7-54

GetPropvalue Reporter 9-77

GetSolutionVariation 14-34

GetTopDesignList 6-42

GetVariables 2-75, 6-43, 7-59

GetVariableValue 2-75, 6-42, 7-58

InsertDesign 6-45, 7-66

LoadSolution 14-38

Paste 6-48, 6-48

Redo 6-48

RemoveAllUnusedDefinitions 6-49

RemoveMaterial 6-49

RemoveUnusedDefinitions 6-50

Rename 6-51

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RestoreProjectArchive 4-44
Save 6-52
SaveAs 6-53
SaveAsStandAloneProject 6-55
SaveProjectArchive 6-56
SetActiveDesign 6-57
SetPropertyValue 2-76, 6-59, 7-74
SetPropValue Design 7-73
SetPropValue Optimetrics 11-83, 11-142
SetPropValue Project 6-58, 9-104
SetVariableValue 2-77, 6-60, 7-75
SimulateAll 6-8, 6-61
Undo 6-61
UpdateDefinitions 6-62
Project Object Script Commands 6-1
property commands
GetArrayVariables 2-72, 6-28
GetProperties 2-72, 6-39, 7-54
GetPropertyValue 2-73, 6-40, 7-55, 9-75
GetPropEvaluatedValue 6-36, 7-51, 11-40, 11-114
GetPropSIValue 6-38, 7-53, 11-42, 11-115
GetVariables 2-75, 6-43, 7-59
GetVariableValue 2-75, 6-42, 7-58
SetPropertyValue 2-76, 6-59, 7-74
SetVariableValue 2-77, 6-60, 7-75
Property Method List 16-82
Property Script Commands 2-25
Conventions uSed in thie Chapter 2-36
Object Script Property Function Summary 2-27
PropertyExists 2-67
PropHost Functions 2-48
PushExcitations 16-68
Layout Editor 16-68

Q

QuitApplication 4-42

R

Radiation module commands
GetSetupNames 10-26
Redo 17-18
design-level command 7-69
project-level command 6-48
RefreshJobMonitor 4-42
RefreshMeshOverlays 17-169
ReleaseData 9-93
Remove 17-33
RemoveAllUnusedDefinitions 6-49
RemoveAnalysisOptions 17-177
RemoveImportData 7-69
RemoveMaterial 6-49
RemoveProp 2-67

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

RemoveSimulationSetup 17-177
RemoveUnusedDefinitions 6-50
Rename 6-51, 14-39, 17-33
RenameAnalysisOptions 17-178
RenameDesignInstance 7-70, 17-19
RenameDrivenSweep 10-62
RenameImportData 7-70, 17-19
RenameReport 9-101
RenameSetup 10-63
Analysis module command 10-63
Optimetrics module command 11-82, 11-141
RenameSimulationSetup 17-178
RenameSource 7-71, 17-184
RenameSweep 17-169
RenameTraces 9-101
Renormalize 14-40
Reorder 14-41, 14-42
Report module commands
GetChildNames 9-69
Reporter editor commands
AddAllEyeMeasurements 9-4
AddCartesianLimitLine 9-4
AddCartesianLimitLineFromCurve 9-5
AddCartesianLimitLineFromEquation 9-7
AddCartesianXMarker 9-8, 9-9
AddCartesianYMarkerToStack 9-10
AddDeltaMarker 9-10
AddMarker 9-11
AddNote 9-12
AddQuickEyeAnalysis 17-79
AddTraceCharacteristics 9-14
AddTraces 9-15
AddVerifyEyeAnalysis 17-94
ApplyReportTemplate 9-17
ChangeProperty 9-17
ClearAllMarkers 9-24
ClearAllTraceCharacteristics 9-24
CloneReportsFromDatasetSolution 9-25
CopyEyeItemAsCommand 7-17, 17-3
CopyPlotSettings 9-26
CopyReportDefinition 9-26
CopyReportsData 9-27
CopyTraceDefinitions 9-27
CopyTracesData 9-28
CreateReport 7-18, 9-29
CreateReportFromTemplate 9-41
CreateReportOfAllQuantities 9-42
DeleteAllReports 9-43
DeleteReport 9-44
DeleteTraceCharacteristics 9-44
DeleteTraces 9-45

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

DoesSupportTraceCharacteristics 9-46
DumpAllReportsData 9-47
EditCartesianXMarker 9-47
EditCartesianYMarker 9-48
EditMarker 9-49
EditQuickEyeAnalysis 17-147
EditVerifEyeAnalysis 17-162
ExportEyeMaskViolation 9-49
ExportImageToFile 9-50
ExportPlot3DToFile 9-58
ExportReport 7-41, 9-59
ExportReportDataToFile 9-60
ExportTableToFile 9-60
ExportToFile 9-62, 9-94
ExportUniformPointsToFile 9-63
GetAllCategories 9-65
GetAllQuantities 9-66
GetAllReportNames 9-66
GetAvailableDisplayTypes 9-67
GetAvailableReportTypes 9-67
GetAvailableSolutionss 9-68
GetChildTypes ReportSetup 9-70
GetCurvePropServerName 9-70
GetDisplayType 9-71
GetDynLinkIntrinsicVariables 9-72
GetDynLinkQtyValueState 9-72
GetDynLinkTraces 9-73
GetDynLinkVariableValues 9-73
GetPropNames 9-77
GetQtyExpressionsForSourceTrace 9-78
GetReportSummaryForRegressionTesting 9-79
GetReportTraceNames 9-78
GetSolutionContexts 9-79, 9-80
GroupPlotCurvesByGroupingStrategy 9-93
ImportReportDataIntoReport 9-95
MovePlotCurvestoGroup 9-95
MovePlotCurvestoNewGroup 9-96
OpenWindowForAllReports 9-97
OpenWindowForReports 9-97
PastePlotSettings 9-98
PasteReports 9-98
PasteReportsWithLegacyNames 9-99
PasteTraces 9-100
PasteTracesWithLegacyNames 9-100
RenameReport 9-101
RenameTraces 9-101
ResetPlotSettings 9-102
SavePlotSettingsAsDefault 9-103
SetLinkOutputTraces 9-103
UnGroupPlotCurvesInGroup 9-105
UpdateAllReports 9-105
UpdateReports 9-106
UpdateTraces 9-106

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateTracesContextAndSweeps 9-109
Reporter Editor Script Commands 9-1
Reporter module commands
GetChildObject 9-69
Reset 14-43
ResetLogging 4-43
ResetPlotSettings 9-102
RestoreWindow 4-45
RestToTimeZero 10-63
ResumeRecording 4-45
RevertAllToInitial 10-64
RevertAllToZeroDisplacement 10-64
RevertSetupToInitial 10-65
RevertSetupToZeroDisplacement 10-65
Rotate 16-70
Running Instance Manager Script Commands 5-1
RunProgram 4-46
RunScript 4-47
RunScriptWithArguments 4-48
RunToolkit 6-52, 7-72

S

Save 6-52
SaveAs 6-53
SaveAsStandAloneProject 6-55
SavePlotSettingsAsDefault 9-103
Schematic Scripting 16-1
Create Method List 16-5
script commands
general method script commands 16-33
scripting
object-oriented 2-1
scripts
CPython 1-44
IronPython 1-2
overview 1-1
pausing 1-53
recording 1-54
running 1-53
stopping 1-53
Scripts and Locked Layers 1-60
SelectAll 16-72
SelectPage 16-73
SelectScheduler 4-13, 4-49
SendToBack 16-73
sensitivity commands
EditSetup 11-25, 11-159
InsertSetup 11-77, 11-163
Sensitivity Script Commands 11-159
SetActiveDefinitionEditor 6-57
SetActiveDesign 6-57
SetActiveEditor 7-72, 17-20
SetActiveProject 4-51
SetActiveProjectByPath 4-51

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

SetAllPortImpedance 14-44
SetCallback 2-68
SetDescription 2-69
SetHidden 2-69
SetLibraryDirectory 4-54
SetLinkOutputTraces 9-103
SetPageData 16-74
SetPortDeembedDistance 14-45
SetPortImpedance 14-46
SetPostProcSettings 14-47
SetProjectDirectory 4-54
SetPropertyValue 2-76, 6-59, 7-74
SetPropValue Design 7-73
SetPropValue Optimetrics 11-83, 11-142
SetPropValue Project 6-58, 9-104
SetReadOnly 2-70
SetTempDirectory 4-57
SetText 2-70
Setting Numerical Values 1-59
SetValue 2-71
SetVariableValue 2-77, 6-60, 7-75
SGetAppDesktop 3-1
ShowVariableBlock 16-76
SimulateAll 6-8, 6-61
SimulateLink 7-76
SimValueContext 8-9
Sleep 4-59
Smooth 14-48
Solutions module commands
DeleteSolutionVariation 7-33
ExportForSpice 7-39
SolveAllSetup, Optimetrics module command 11-84, 11-143
SolveSetup 10-66
SolveSetup, Optimetrics module command 11-84, 11-143
SortComponents 16-77
statistical commands
EditSetup 11-28, 11-166
InsertSetup 11-79, 11-167
Statistical Script Commands 11-166
Stretch 14-49
Structures 1-30
SubmitJob 4-59

T

Terminate 14-50

U

UDM Structures 1-30
Undo 17-21
project-level command 6-61
UnGroupPlotCurvesInGroup 9-105
UpdateAllReports 9-105
UpdateDefinitions, project-level command 6-62
UpdateReports 9-106
UpdateTraces 9-106

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.

UpdateTracesContextAndSweeps 9-109

UseCircuitSParameterDefinition 7-77, 17-21

User defined documents module commands

GetDocumentNames 12-5

User defined solution module commands

CreateUserDefinedSolution 13-1

DeleteUserDefinedSolutions 13-2

EditUserDefinedSolution 13-3

GetUserDefinedSolutionNames 13-4

GetUserDefinedSolutionProperties 13-4

User Defined Solutions Commands 13-1

V

variables

hierarchy in HFSS 1-50

used in HFSS scripts 1-50

W

Wire (Schematic Editor) 16-78

Z

ZoomArea 16-78

ZoomIn 16-79

ZoomOut 16-80

ZoomPrevious 16-81

ZoomToFit 16-81

Ansys Electromagnetics Suite 2026 R1 - © ANSYS, Inc. All rights reserved. - Contains proprietary and confidential information of ANSYS, Inc. and its subsidiaries and affiliates.