



SIEMENS EDA

Variant Manager Automation Reference

Release X-ENTP VX.2.13
Document Revision 10

Unpublished work. © 2023 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer's applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/base/uca/>, as supplemented by the product specific terms which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/supplements/>.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' Marks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a leading global provider of product life cycle management (PLM) software and services with 7 million licensed seats and 71,000 customers worldwide. Headquartered in Plano, Texas, Siemens Digital Industries Software works collaboratively with companies to deliver open solutions that help them turn more ideas into successful products. For more information on Siemens Digital Industries Software products and services, visit www.siemens.com/plm.

Support Center: support.sw.siemens.com

Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Revision History ISO-26262

Revision	Changes	Status/Date
10	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. No technical enhancements, changes, and fixes listed in the <i>Xpedition Enterprise Flow Release Notes</i> for this product are reflected in this document. Approved by Mike Bare.	Released March 2023
9	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. No technical enhancements, changes, and fixes listed in the <i>Xpedition Enterprise Flow Release Notes</i> for this product are reflected in this document. Approved by Mike Bare.	Released September 2022
8	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the <i>Xpedition Enterprise Flow Release Notes</i> for this product are reflected in this document. Approved by Mike Bare.	Released March 2022
7	Modifications to title page to reflect the latest product version supported. Approved by Kevin Chupp. All technical enhancements, changes, and fixes listed in the <i>Xpedition Enterprise Flow Release Notes</i> for this product are reflected in this document. Approved by Mike Bare.	Released September 2021

Author: In-house procedures and working practices require multiple authors for documents. All associated authors for each topic within this document are tracked within the Siemens documentation source. For specific topic authors, contact the Siemens Digital Industries Software documentation department.

Revision History: Released documents include a revision history of up to four revisions. For earlier revision history, refer to earlier releases of documentation on Support Center.

Table of Contents

Chapter 1

Variant Manager Automation.....	11
Overview.....	11
Learning More About Automation.....	11
Variant Manager Automation Data Model.....	11
Accessing Variant Manager Automation.....	13
Using the Variant Manager Addin Control.....	13
Using VariantManagerControl from Layout Applications.....	15

Chapter 2

Variant Manager Objects.....	19
Application Object.....	21
BOM Property (Application Object).....	23
CorporateSettings Method (Application Object).....	24
FXEDData Property (Application Object).....	25
GenerateFunctionView Method (Application Object).....	26
GenerateVariantView Method (Application Object).....	27
IsReloadingNeeded Property (Application Object).....	28
IsVariantView Method (Application Object).....	29
LoadSettings Method (Application Object).....	30
ReadOnly Property (Application Object).....	31
ReConnectHost Method (Application Object).....	32
RecreateFMVVariants Method (Application Object).....	33
ResetView Method (Application Object).....	34
Save Method (Application Object).....	35
Settings Property (Application Object).....	36
VMDocument Property (Application Object).....	37
OnBlockModificationChanged Event (Application Object).....	38
OnComponentModificationChanged Event (Application Object).....	39
OnDataChanged Event (Application Object).....	40
OnFunctionChanged Event (Application Object).....	41
OnFunctionGroupChanged Event (Application Object).....	42
OnMatrixChanged Event (Application Object).....	43
OnSymbolModificationChanged Event (Application Object).....	44
OnVariantChanged Event (Application Object).....	45
OnVariantGroupChanged Event (Application Object).....	46
VMBlockModification Object.....	47
BlockSymbol Property (VMBlockModification Object).....	48
ChangeOperation Method (VMBlockModification Object).....	49
Delete Method (VMBlockModification Object).....	50
Operation Property (VMBlockModification Object).....	51
SubVariantName Property (VMBlockModification Object).....	52
variant Property (VMBlockModification Object).....	53
VMBorderPropModification Object.....	54

Table of Contents

BorderSymbol Property (VMBorderPropModification Object).....	55
Delete Method (VMBorderPropModification Object).....	56
Name Property (VMBorderPropModification Object).....	57
value Property (VMBorderPropModification Object).....	58
variant Property (VMBorderPropModification Object).....	59
VMBOM Object.....	60
Add Method (VMBOM Object).....	61
ClearList Method (VMBOM Object).....	62
Execute Method (VMBOM Object).....	63
Format Property (VMBOM Object).....	64
GenerateIndex Property (VMBOM Object).....	65
GenerateReuseBlocksReports Property (VMBOM Object).....	66
IncludeMaster Property (VMBOM Object).....	67
OutputPath Property (VMBOM Object).....	68
Properties Property (VMBOM Object).....	69
Type Property (VMBOM Object).....	70
UserProps Property (VMBOM Object).....	71
Variants Property (VMBOM Object).....	72
VMColor Object.....	73
blue Property (VMColor Object).....	74
green Property (VMColor Object).....	75
red Property (VMColor Object).....	76
VMComponent Object.....	77
AliasPartNumber Property (VMComponent Object).....	78
ContainsNested Property (VMComponent Object).....	79
Name Property (VMComponent Object).....	80
OriginalAliasPartNumber Property (VMComponent Object).....	81
OriginalPartNumber Property (VMComponent Object).....	82
ParentComponent Property (VMComponent Object).....	83
PartNumber Property (VMComponent Object).....	84
SID Property (VMComponent Object).....	85
Symbols Property (VMComponent Object).....	86
Type Property (VMComponent Object).....	87
variant Property (VMComponent Object).....	88
VMComponentModification Object.....	89
ChangeOperation Method (VMComponentModification Object).....	90
ChangeOperationEx Method (VMComponentModification Object).....	91
Component Property (VMComponentModification Object).....	92
Delete Method (VMComponentModification Object).....	93
newAliasPartNumber Property (VMComponentModification Object).....	94
NewPartNumber Property (VMComponentModification Object).....	95
Operation Property (VMComponentModification Object).....	96
variant Property (VMComponentModification Object).....	97
VMDocument Object.....	98
AssignFunctionToVariant Method (VMDocument Object).....	100
BlockModifications Property (VMDocument Object).....	101
BorderPropModifications Property (VMDocument Object).....	102
ComponentModifications Property (VMDocument Object).....	103
Components Property (VMDocument Object).....	104

DeleteBlockModification Method (VMDocument Object).....	105
DeleteBorderPropModification Method (VMDocument Object).....	106
DeleteComponentModification Method (VMDocument Object).....	107
DeleteFunction Method (VMDocument Object).....	108
DeleteFunctionGroup Method (VMDocument Object).....	109
DeleteSymbolModification Method (VMDocument Object).....	110
DeleteVariant Method (VMDocument Object).....	111
DeleteVariantGroup Method (VMDocument Object).....	112
FindBlockModification Method (VMDocument Object).....	113
FindBorderPropModification Method (VMDocument Object).....	114
FindComponent Method (VMDocument Object).....	115
FindComponentModification Method (VMDocument Object).....	116
FindFunction Method (VMDocument Object).....	117
FindFunctionGroup Method (VMDocument Object).....	118
FindSymbol Method (VMDocument Object).....	119
FindSymbolModification Method (VMDocument Object).....	120
FindVariant Method (VMDocument Object).....	121
FindVariantGroup Method (VMDocument Object).....	122
FunctionGroups Property (VMDocument Object).....	123
Functions Property (VMDocument Object).....	124
GetBorderSymbols Method (VMDocument Object).....	125
LockRedraw Method (VMDocument Object).....	126
PutBlockModification Method (VMDocument Object).....	127
PutBorderPropModification Method (VMDocument Object).....	128
PutComponentModification Method (VMDocument Object).....	129
PutFunction Method (VMDocument Object).....	130
PutFunctionGroup Method (VMDocument Object).....	131
PutSymbolModification Method (VMDocument Object).....	132
PutVariant Method (VMDocument Object).....	133
PutVariantGroup Method (VMDocument Object).....	134
SelectVariantGroup Method (VMDocument Object).....	135
SymbolModifications Property (VMDocument Object).....	136
Symbols Property (VMDocument Object).....	137
UnAssignFunctionToVariant Method (VMDocument Object).....	138
UnLockRedraw Method (VMDocument Object).....	139
VariantGroups Property (VMDocument Object).....	140
Variants Property (VMDocument Object).....	141
VMFunctionGroup Object.....	142
Assign Method (VMFunctionGroup Object).....	143
AssignAll Method (VMFunctionGroup Object).....	144
Description Property (VMFunctionGroup Object).....	145
Functions Property (VMFunctionGroup Object).....	146
Name Property (VMFunctionGroup Object).....	147
UnAssign Method (VMFunctionGroup Object).....	148
UnAssignAll Method (VMFunctionGroup Object).....	149
VMFunction Object.....	150
Description Property (VMFunction Object).....	151
Name Property (VMFunction Object).....	152
VMFXEDData Object.....	153

Table of Contents

Add Method (VMFXEDData Object).....	154
ClearList Method (VMFXEDData Object).....	155
Execute Method (VMFXEDData Object).....	156
Variants Property (VMFXEDData Object).....	157
VMGUIProperty Object.....	158
Delete Method (VMGUIProperty Object).....	159
Name Property (VMGUIProperty Object).....	160
value Property (VMGUIProperty Object).....	161
VMSettingsCAE Object.....	162
GetMarkupColor Method (VMSettingsCAE Object).....	163
GetUnplaceColor Method (VMSettingsCAE Object).....	164
LoadSchematicOnlyComponents Property (VMSettingsCAE Object).....	165
MarkupColor Property (VMSettingsCAE Object).....	166
PackageUnplaceOption Property (VMSettingsCAE Object).....	167
RemoveDanglingWires Property (VMSettingsCAE Object).....	168
SetMarkupColor Method (VMSettingsCAE Object).....	169
SetUnplaceColor Method (VMSettingsCAE Object).....	170
ShowSymbolReplacements Property (VMSettingsCAE Object).....	171
UnplaceColor Property (VMSettingsCAE Object).....	172
VMSettingsGeneral Object.....	173
CentLibPropsInGrid Property (VMSettingsGeneral Object).....	174
DisplaySelectedPropsOnly Property (VMSettingsGeneral Object).....	175
ReportTxtDelimiter Property (VMSettingsGeneral Object).....	176
ReportViewExcel Property (VMSettingsGeneral Object).....	177
ReportViewHTML Property (VMSettingsGeneral Object).....	178
UnplacedKeyword Property (VMSettingsGeneral Object).....	179
VMSettingsLibQuery Object.....	180
AliasPartNumberAttribute Property (VMSettingsLibQuery Object).....	181
ConfigFile Property (VMSettingsLibQuery Object).....	182
MaxResultsPerLib Property (VMSettingsLibQuery Object).....	183
MaxResultsTotal Property (VMSettingsLibQuery Object).....	184
PartNumberAttribute Property (VMSettingsLibQuery Object).....	185
RowFormatter Property (VMSettingsLibQuery Object).....	186
SummaryInfoFormatter Property (VMSettingsLibQuery Object).....	187
VMSettingsPADS Object.....	188
AttributePartName Property (VMSettingsPADS Object).....	189
VMSettingsPCB Object.....	190
DisplayNestedMechanicalCells Property (VMSettingsPCB Object).....	191
FablinkActionPrompt Property (VMSettingsPCB Object).....	192
FablinkDataGenerationMode Property (VMSettingsPCB Object).....	193
RemoveNested Property (VMSettingsPCB Object).....	194
UnplaceAssKeepOutline Property (VMSettingsPCB Object).....	195
UnplaceAssKeepPartnum Property (VMSettingsPCB Object).....	196
UnplaceAssKeepRefdes Property (VMSettingsPCB Object).....	197
UnplaceSilkKeepOutline Property (VMSettingsPCB Object).....	198
UnplaceSilkKeepPartnum Property (VMSettingsPCB Object).....	199
UnplaceSilkKeepRefdes Property (VMSettingsPCB Object).....	200
UnplaceSolderMask Property (VMSettingsPCB Object).....	201
UnplaceSolderPaste Property (VMSettingsPCB Object).....	202

VMSettings Object.....	203
CAE Property (VMSettings Object).....	205
General Property (VMSettings Object).....	206
LibQuery Property (VMSettings Object).....	207
PADS Property (VMSettings Object).....	208
PCB Property (VMSettings Object).....	209
VMSymbolModification Object.....	210
ChangeOperation Method (VMSymbolModification Object).....	211
Delete Method (VMSymbolModification Object).....	212
function Property (VMSymbolModification Object).....	213
NewPartNumber Property (VMSymbolModification Object).....	214
Operation Property (VMSymbolModification Object).....	215
symbol Property (VMSymbolModification Object).....	216
VMSymbol Object.....	217
Component Property (VMSymbol Object).....	218
Name Property (VMSymbol Object).....	219
NameAndPath Property (VMSymbol Object).....	220
ParentSymbol Property (VMSymbol Object).....	221
Symbols Property (VMSymbol Object).....	222
Type Property (VMSymbol Object).....	223
VMVariantGroup Object.....	224
Assign Method (VMVariantGroup Object).....	225
AssignAll Method (VMVariantGroup Object).....	226
Description Property (VMVariantGroup Object).....	227
Name Property (VMVariantGroup Object).....	228
UnAssign Method (VMVariantGroup Object).....	229
UnAssignAll Method (VMVariantGroup Object).....	230
Variants Property (VMVariantGroup Object).....	231
VMVariant Object.....	232
BOMInclude Property (VMVariant Object).....	233
BorderProperties Property (VMVariant Object).....	234
DeleteProperty Method (VMVariant Object).....	235
Description Property (VMVariant Object).....	236
FindProperty Method (VMVariant Object).....	237
FXEInclude Property (VMVariant Object).....	238
Hyperlink Property (VMVariant Object).....	239
Name Property (VMVariant Object).....	240
Number Property (VMVariant Object).....	241
PutProperty Method (VMVariant Object).....	242
Type Property (VMVariant Object).....	243

Chapter 3

Variant Manager Collections..... 245

VMBlockModifications Collection.....	246
Count Property (VMBlockModifications Collection).....	247
Item Property (VMBlockModifications Collection).....	248
VMBorderPropModifications Collection.....	249
Count Property (VMBorderPropModifications Collection).....	250

Table of Contents

Item Property (VMBorderPropModifications Collection).....	251
VMComponentModifications Collection.....	252
Count Property (VMComponentModifications Collection).....	253
Item Property (VMComponentModifications Collection).....	254
VMComponents Collection.....	255
Count Property (VMComponents Collection).....	256
Item Property (VMComponents Collection).....	257
VMFunctionGroups Collection.....	258
Count Property (VMFunctionGroups Collection).....	259
Item Property (VMFunctionGroups Collection).....	260
VMFunctions Collection.....	261
Count Property (VMFunctions Collection).....	262
Item Property (VMFunctions Collection).....	263
VMGUIProperties Collection.....	264
Count Property (VMGUIProperties Collection).....	265
Item Property (VMGUIProperties Collection).....	266
VMSymbolModifications Collection.....	267
Count Property (VMSymbolModifications Collection).....	268
Item Property (VMSymbolModifications Collection).....	269
VMSymbols Collection.....	270
Count Property (VMSymbols Collection).....	271
Item Property (VMSymbols Collection).....	272
VMVariantGroups Collection.....	273
Count Property (VMVariantGroups Collection).....	274
Item Property (VMVariantGroups Collection).....	275
VMVariants Collection.....	276
Count Property (VMVariants Collection).....	277
Item Property (VMVariants Collection).....	278

Chapter 4

Variant Manager Enumerated Types.....279

Variant Manager Enumerated Types Summary.....	280
EVMBOMType Enum.....	281
EVMCAEPackUnplace Enum.....	282
EVMComponentType Enum.....	283
EVMErrCode Enum.....	284
EVMEvOperation Enum.....	286
EVMFabLinkDataGenerationMode Enum.....	287
EVFMVOperation Enum.....	288
EVMGRGUICAEPackUnplace Enum.....	289
EVMOperation Enum.....	290
EVMOptionOnOffToggle Enum.....	291
EVMPADSOutputErrorCode Enum.....	292
EVMReportFormat Enum.....	293
EVMSymbolType Enum.....	294
EVMVariantType Enum.....	295

Third-Party Information

Chapter 1

Variant Manager Automation

This document provides reference information for the supported Automation objects, object collections, and enumerated types for Variant Manager

[Overview](#)

[Learning More About Automation](#)

[Variant Manager Automation Data Model](#)

[Accessing Variant Manager Automation](#)

Overview

This section provides information you need to get started with Variant Manager Automation.

Variant Manager Automation is a COM (Component Object Model) - based interface that allows you to create scripts that read and write Variant Manager data. Using Automation scripts you can:

- Create custom application setups
- Create default Variant Manager design data, such as variants, variant groups, functions and function groups
- Perform required assignments
- Perform custom checks that integrate with the logical or layout design

Several Mentor Graphics PCB applications support Variant Manager and its automation interface.

To use Variant Manager Automation, you must have Variant Manager licensing.

Learning More About Automation

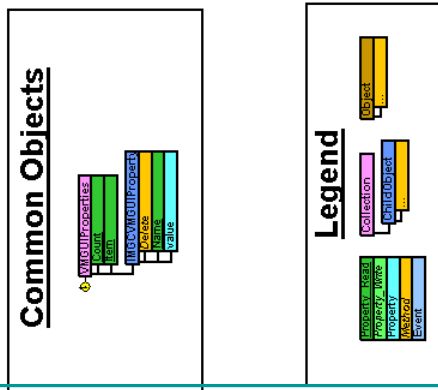
If you are new to Automation, the *Layout Automation Tutorial* is available for download from the Mentor Graphics Support Center website. The tutorial includes exercises with step by step procedures and discussion for creating scripts in the Mentor Graphics PCB environment.

There are a myriad of books and web sites dedicated to helping you learn good practices and techniques for script programming. The resources you choose to help you learn are largely a function of the scripting language you are using and the platform from which you operate. Rather than attempt to list all these resources in this space, Mentor Graphics recommends that you search the internet for an extensive list of the many resources available.

Variant Manager Automation Data Model

The following figure shows the data model for the Variant Manager Automation.

Variant Manager Automation Reference



Accessing Variant Manager Automation

There are a various methods of accessing the Variant Manager Automation. How you do this depends largely on the usage model.

The code snippets in this section show some of the more common methods.



CAUTION:

When using Automation to modify Variant Manager objects or data, you should not use two or more levels of indirection in a single statement. Unpredictable behavior may occur.

```
'This statement contains two levels of indirection.  
Set var = vmgr.VMDocument.FindVariant(sVariantName)  
  
'Separate into two statements, each with one level of indirection.  
Set vmDoc = vmgr.VMDocument  
Set var = vmDoc.FindVariant(sVariantName)
```

[Using the Variant Manager Addin Control](#)

[Using VariantManagerControl from Layout Applications](#)

Using the Variant Manager Addin Control

The following shows an example of opening Variant Manager, which is an addin, using the Addins collection of Mentor Graphics PCB applications. Using this example, you can write a single script that utilizes Variant Manager in all PCB applications.



Note:

With the introduction of VX.1, you must use the following library for any scripts that invoke the MGCVariantGUICtrl library: **MGCVariantGUICtrl.<version>**

Example 1. Variant Manager Addin

```
Option Explicit  
  
Dim addinName : addinName = "Variant Manager"  
Dim visible : visible = True  
Dim sComVersion  
sComVersion = GetProgIDVer()  
Dim progID : progID = "MGCVARIANTGUI.MGCVariantGUICtrl." + sComVersion  
Dim associatedScript : associatedScript = ""  
Dim addinPlacement : addinPlacement = 0  
Dim shortcutKey : shortcutKey = ""  
Dim addinObj  
Dim docObj  
Dim appObj
```

```
Set addinObj = launchAddin(progID, addinName, associatedScript, _
    addinPlacement, shortcutKey)
If Not (addinObj Is Nothing) Then
    Scripting.AddTypeLibrary(progID)
    ' MsgBox "EXP_PROG_ID_VER = " & sComVersion
    addinObj.Visible = visible
    Set appObj = addinObj.Control.VariantGUIApplication
    Set docObj = appObj.VMDocument
    ' INITIALIZATION CODE GOES HERE
Else
    MsgBox "No Addin"
End If

Function launchAddin(progID, localName, associatedScript, _
    defaultPlacement, shortcutKey)
    ' launch an Addin if it's not already running

    Dim toolBarMenu: toolBarMenu = False
    Dim quiet: quiet = True
    Dim groupName: groupName = ""

    Set launchAddin = alreadyLoaded(localName)
    If launchAddin Is Nothing Then
        MsgBox "Addin not loaded"
        ' add it to the addins collection, which will launch it
        Set launchAddin = Addins.Add(progID, localName, associatedScript, _
            defaultPlacement, shortcutKey, groupName, quiet, toolbarMenu)
        If launchAddin Is Nothing Then
            MsgBox "Addin not created."
        End If
    End If
End Function

Function alreadyLoaded(localName)
    ' is the named addin already loaded? Iterate collection
    ' returns Nothing if addin is not found, returns addin if found

    Set alreadyLoaded = Nothing
    Dim ain
    For Each ain In Addins
        If ain.Name = localName Then
            Set alreadyLoaded = ain
            Exit For
        End If
    Next
End Function

' This function returns the progID version of a library
'
Private Function GetProgIDVer()
    On Error Resume Next
    Dim version_string
    version_string = Scripting.GetEnvVariable("EXP_PROG_ID_VER")
    If version_string = "" Then
```

```
version_string = Scripting.GetEnvVariable("DXD_PROG_ID_VER")
End If
If version_string = "" Then
    version_string = "1"
End If
GetProgIDVer = version_string
End Function
```

Referenced Constructs

Noteworthy objects, properties and methods in this example include:

- The Scripting Object, includes the AddTypeLibrary method.
- The Addins Collection, includes the Add method, Name and ProgID properties.

Using VariantManagerControl from Layout Applications

The following shows an example of opening Variant Manager using the Document VariantManagerControl property of the MGCPB server.

(See Document.VariantManagerControl for more information.).

Using this property requires that the application already have Variant Manager loaded in the application's addin collection. The snippet accesses Variant Manager from Expedition PCB. It iterates the component modifications, selecting and counting the unplaced components. The resulting count displays in the status bar. The view in Expedition, zooms to the selected (unplaced) components.

Example 2. Layout Application: VariantManagerControl Property

```
Sub Document_VariantManager()

    Dim docObj,resultMess,VMapObj
    Dim compModsColl, compModObj, totalUnplacedComps, pcbCompObj

    ' collect PCB document object
    Set docObj = GetLicensedDoc(app)
    If (docObj Is Nothing) Then Exit Sub

    ' Get the VM application
    Set VMapObj = _
        docObj.VariantManagerControl.Control.VariantGUIApplication
    If (Err) Then
        Call app.Gui.StatusBarText("An error occurred while getting VM: " + _
            vbNewLine + Err.Description,epcbStatusFieldError)
        Exit Sub
    Else
        totalUnplacedComps = 0
        docObj.UnselectAll
        Set compModsColl = _
            VMapObj.VMDocument.ComponentModifications("Variant1")
```



```
' Iterate the collection, check if unplaced component
For Each compModObj In compModsColl
  If (compModObj.Operation = eVMOperUnplace ) Then
    ' Find the component in the PCB database and select it
    Set pcbCompObj = _
      docObj.FindComponent(compModObj.Component.Name)
    If Not pcbCompObj Is Nothing Then
      pcbCompObj.Selected = True
    End If
    ' Increment the count of unplaced components
    totalUnplacedComps = totalUnplacedComps + 1
  End if
Next
resultMess = "There are " + CStr(totalUnplacedComps) + _
  " unplaced components in the variant design"
' select the unplaced components and fit it on the board
docObj.ActiveView.SetExtentsToSelection

' Update status bar
Call app.Gui.StatusBarText(resultMess,epcbStatusField1)
End If
End Sub

' This function returns the progID version of a library
,
Private Function GetProgIDVer()
  On Error Resume Next
  Dim version_string
  version_string = Scripting.GetEnvVariable("EXP_PROG_ID_VER")
  If version_string = "" Then
    version_string = Scripting.GetEnvVariable("DXD_PROG_ID_VER")
  End If
  If version_string = "" Then
    version_string = "1"
  End If
  GetProgIDVer = version_string
End Function

' This function retrieves an automation license for a document
,
' Returns:
' - 'Nothing' if licensing failed .
' - A reference to the active document of the application if
'   licensing succeeded.
,
Public Function GetLicensedDoc(app)
  On Error Resume next
  Dim key,licenseServer,licenseToken,docObj
  Set GetLicensedDoc = Nothing
  ' collect the active document
  Set docObj = app.ActiveDocument
  If (Err) Then
    Call app.Gui.StatusBarText("No active document: " + _
```

```
    Err.Description,epcbStatusFieldError)
Exit Function
End If
' Ask document for the key
key = docObj.Validate(0)
' Get token from license server
Set licenseServer = _
    CreateObject("MGCPBAutomationLicensing.Application")
licenseToken = licenseServer.GetToken(key)
Set licenseServer = Nothing
' Ask the document to validate the license token
Err.Clear
docObj.Validate(licenseToken)
If (Err) Then
    Call app.Gui.StatusBarText("No active document license: " + _
        Err.Description,epcbStatusFieldError)
    Exit Function
End If
' everything is OK, return document
Set GetLicensedDoc = docObj
End Function
'
' Testing code for the sample
'

Dim app
Set app = GetObject(,"MGCPCB.Application")
Dim sComVersion
sComVersion = GetProgIDVer()
Dim progID : progID = "MGCVARIANTGUI.MGCVariantGUICtrl." + sComVersion
Scripting.AddTypeLibrary(progID)

Call Document_VariantManager
```

Referenced Constructs

Noteworthy objects, properties and methods in this example include:

- The Document.VariantManagerControl property (part of MGCPCB Automation).

Chapter 2

Variant Manager Objects

This section contains an alphabetical listing of Variant Manager Automation Objects.

[Table 1: Variant Manager Objects](#) includes summary information for each object you can access in Variant Manager Automation. To view the full description for a specific object, click the Object name.

Table 1. Variant Manager Objects

Object	Description
Application Object	The Application object represents the Variant Manager application. This is the root level object. Use the object to access the VMDocument object (the design), to work with application settings, to create variant views, and to create reports.
VMBlockModification Object	The VMBlockModification object represents a modified block. Use this object to access the modified block, its variant and the modification operation. You can also set the operation type for modifying blocks.
VMBorderPropModification Object	The VMBorderPropModification object represents a modified border property. Use this object to access the modified border property, its variant and the modification operation.
VMBOM Object	The VMBOM object represents the BOM Reports dialog. Use this object to define the format, settings and variants to report in a BOM.
VMColor Object	The VMColor object represent the red, green, blue values for a color object.
VMComponent Object	The VMComponent object represents a physical component. Use the object to access attributes of the component including its name, current and original part numbers, the symbols that define the component and the variant associated with the component.
VMComponentModification Object	The VMComponentModification object represents a modified component. Use this object to access the modified component, its variant and the modification operation. You can also set the operation type for modifying components.
VMDocument Object	The VMDocument object represent a Variant Manager design. Use this object to delete, find and add Variant Manager design objects (Components, Symbols, Functions, Variants, Function Groups, Variant Groups, Component Modifications and Symbol Modifications).
VMFunctionGroup Object	The VMFunctionGroup object represents a function group. Use this object to assign and unassign functions to a function group and to access the name, description and functions of the function group.
VMFunction Object	The VMFunction object represents a function. Use this object to access the name and description of a function.
VMFXEDData Object	The VMFXEDData object represent the Xpedition FabLink data. Use the object to add (or clear) variants or variant groups to the assigned list, or to access the variants currently assigned to the list. This object also performs the data generation.
VMGUIProperty Object	The VMGUIProperty object represents a name/value property pair. Use this object to access properties.

Table 1. Variant Manager Objects (continued)

Object	Description
VMSettingsCAE Object	The VMSettingsCAE object represents the CAE Interaction tab of the Settings dialog. Use this object to customize parameters when Variant Manager is open in the schematic design application.
VMSettingsGeneral Object	The VMSettingsGeneral object represents the General tab of the Setting dialog. Use this object to define the common settings for all Variant Manager clients.
VMSettingsLibQuery Object	The VMSettingsLibQuery object represents the Library Query Setup tab of the Settings dialog. Use this object to setup the connection parameters for the library data source.
VMSettingsPADS Object	The VMSettingsPADS object represents the PADS tab of the Settings dialog. Use this object to setup the PADS options.
VMSettingsPCB Object	The VMSettingsPCB object represents the PCB Interaction tab of the Settings dialog. Use this object to define the settings specific to the PCB layout and manufacturing applications.
VMSettings Object	The VMSettings object represents the Settings dialog. Use this object to return the settings objects for General, PCB Interaction, CAE Interaction, PADS Interaction and Library Query Setup.
VMSymbolModification Object	The VMSymbolModification object represents a modified symbol. Use this object to access the modified symbol, its function and the modification operation. You can also set the operation type for modifying symbols.
VMSymbol Object	The VMSymbol object represents a logic symbol. Use this object to access symbol attributes including, name, unique name (identifier), physical component, and symbol type. For hierarchical symbols, the object also accesses the parent symbol and the sub-symbols.
VMVariantGroup Object	The VMVariantGroup object represents a variant group. Use this object to assign and unassign variants to a variant group and to access the name, description and variants of the variant group.
VMVariant Object	The VMVariant object represent a physical variant or a function managed variant (FMV). Use this object to access the attributes of a variant including the name, field number, description and variant type (physical or FMV). You can also work with the border properties, and set the options to include the variant in BOM reports and Xpedition FabLink data.

Application Object

The Application object represents the Variant Manager application. This is the root level object. Use the object to access the VMDocument object (the design), to work with application settings, to create variant views, and to create reports.

The following section defines the methods, properties and events of the Application object.

Table 2. Application Object Methods, Properties, and Events

Method, Property, or Event	Description
BOM Property (Application Object)	Returns the VMBOM object representing BOM generation methods and properties.
CorporateSettings Method (Application Object)	Opens the Import Corporate Settings dialog box.
FXEDData Property (Application Object)	Returns the FXEDData object representing data generation for Xpedition FabLink.
GenerateFunctionView Method (Application Object)	Generates function view.
GenerateVariantView Method (Application Object)	Generates variant view.
IsReloadingNeeded Property (Application Object)	Returns the status of the reloading needed flag.
IsVariantView Method (Application Object)	Checks if the main application is variant (or function) view.
LoadSettings Method (Application Object)	Loads the settings for the editor from the given path.
ReadOnly Property (Application Object)	Returns the value of the read-only flag.
ReConnectHost Method (Application Object)	Reconnect a host to the application.
RecreateFMVVariants Method (Application Object)	Recreates or regenerates FMV variants after modification. Regenerates FMVs after a function or function matrix assignment is modified.
ResetView Method (Application Object)	Resets view to master view.
Save Method (Application Object)	Saves all data and settings in Variant Manager.
Settings Property (Application Object)	Returns the settings object.
VMDocument Property (Application Object)	Returns the VM document (design).
OnBlockModificationChanged Event (Application Object)	Fired when something has changed in the block modification.
OnComponentModificationChanged Event (Application Object)	Fires when something has changed in the component assignment.
OnDataChanged Event (Application Object)	Fires when something has changed in the document.
OnFunctionChanged Event (Application Object)	Fires when something has changed in the function.

Table 2. Application Object Methods, Properties, and Events (continued)

Method, Property, or Event	Description
OnFunctionGroupChanged Event (Application Object)	Fires when something has changed in the function group.
OnMatrixChanged Event (Application Object)	Fires when something has changed in variant/function matrix assignment.
OnSymbolModificationChanged Event (Application Object)	Fires when something has changed in the symbol assignment.
OnVariantChanged Event (Application Object)	Fires when something has changed in the variant.
OnVariantGroupChanged Event (Application Object)	Fires when something has changed in the variant group.

BOM Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the VMBOM object representing BOM generation methods and properties.

Usage

Application.BOM

Arguments

None

Return Values

IMGCVMBOM. The [VMBOM Object](#).

CorporateSettings Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Opens the Import Corporate Settings dialog box.

Usage

Application.CorporateSettings() As Integer

Arguments

None

Return Values

As Integer. Value “1” if you select and import settings; value of “0” if you cancel import settings.

FXEDData Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the FXEDData object representing data generation for Xpedition FabLink.

Usage

Application.FXEDData

Arguments

None

Return Values

IMGCVMFXEDData. The [VMFXEDData Object](#).

GenerateFunctionView Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Generates function view.

Usage

Application.GenerateFunctionView([ByVal function As Variant]) As Boolean

Arguments

- function

(Optional) The function can be either the function name, [VMFunction Object](#), or an empty value. In the case of an empty value, Variant Manager tries to generate a view of the currently active function on the Function tab.

Return Values

As Boolean. "True" if the function view is created successfully; "False" if the function name or object is incorrect or the function view cannot be generated.

GenerateVariantView Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Generates variant view.

Usage

Application.GenerateVariantView([ByVal var As Variant]) As Boolean

Arguments

- var

(Optional) The variant can be either the Variant name, [VMVariant Object](#), or an empty value. In the case of an empty value, Variant Manager tries to generate a view of the currently active variant on the Variants tab.

Return Values

As Boolean. "True" if the variant view is created successfully; "False" if the variant name or object is incorrect or the variant view cannot be generated.

IsReloadingNeeded Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the status of the reloading needed flag.

Usage

Application.IsReloadingNeeded

Arguments

None

Return Values

As Boolean. "True" if variant data needs to be reloaded; "False" if variant data does not need to be reloaded.

IsVariantView Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Checks if the main application is variant (or function) view.

Usage

Application.IsVariantView() As Boolean

Arguments

None

Return Values

As Boolean. "True" if the application is in variant (or function) view; "False" if the application is in master mode.

LoadSettings Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Loads the settings for the editor from the given path.

Usage

```
Application.LoadSettings(fileName As String) As Integer
```

Arguments

- fileName

A string that contains the path to the VmgrCommon.cfg file.

Return Values

As Integer. Value “1” if the path is correct; value “-1” if the path is incorrect.

ReadOnly Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the value of the read-only flag.

Usage

`Application.ReadOnly`

Arguments

None

Return Values

As Boolean. "True" if Variant Manager is open in read-only mode; "False" if Variant Manager is open in normal (full access) mode.

ReConnectHost Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Reconnect a host to the application.



Note:

Executing this method will reload Variant Manager data if the value of the IsReloadingNeeded property is True.

Usage

Application.ReConnectHost() As Integer

Arguments

None

Return Values

As Integer. Value "1" if successful. Value "0" if unsuccessful.

RecreateFMVVariants Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Recreates or regenerates FMV variants after modification. Regenerates FMVs after a function or function matrix assignment is modified.

Usage

Application.RecreateFMVVariants() As Boolean

Arguments

None

Return Values

As Boolean. "True" if the FMV recreates successfully; "False" if the FMV does not recreate successfully.

ResetView Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Resets view to master view.

Usage

Application.ResetView() As Boolean

Arguments

None

Return Values

As Boolean. "True" if the view resets successfully; "False" if the view does not reset successfully.

Save Method (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Saves all data and settings in Variant Manager.

Usage

Application.Save() As Boolean

Arguments

None

Return Values

As Boolean. "True" if the save is successful; "False" if the save is not successful.

Settings Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the settings object.

Usage

Application.Settings

Arguments

None

Return Values

IMGCVMSettings. The [VMSettings Object](#).

VMDocument Property (Application Object)

Object: [Application Object](#)

Access: Read-Only

Prerequisites: None.

Returns the VM document (design).

Usage

Application.VMDocument

Arguments

None

Return Values

IMGCVMDocument. The [VMDocument Object](#).

OnBlockModificationChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fired when something has changed in the block modification.

Usage

```
Sub Application_OnBlockModificationChanged(ByVal symbolName As String,  
    ByVal varName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- symbolName

The string that contains the full name of the modified block.

- varName

The string that contains the name of the variant where the block was modified.

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a block modification is created), eVMOperationDelete (a block modification is deleted), eVMOperationChange (a block modification is changed).

OnComponentModificationChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the component assignment.

Usage

```
Sub Application_OnComponentModificationChanged(ByVal compName As String,  
    ByVal varName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- compName

The string that contains the name of the modified component.

- varName

The string that contains the name of the variant where the component was modified.

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a component modification is created), eVMOperationDelete (a component modification is deleted), eVMOperationChange (a component modification is changed).

OnDataChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the document.

Usage

Sub Application_OnDataChanged()

Arguments

None

OnFunctionChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the function.

Usage

```
Sub Application_OnFunctionChanged(ByVal Name As String,  
    ByVal oldName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- Name

A string that contains the name of the changed function.

- oldName

This argument is valid only for eVMOperationChange. This is a string that contains the old name of the function (if the function name changes) or is an empty string (if the description changes.)

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a new function is created), eVMOperationDelete (an existing function is deleted), or eVMOperationChange (the function name or description changes).

OnFunctionGroupChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the function group.

Usage

```
Sub Application_OnFunctionGroupChanged(ByVal fnGrpName As String,  
    ByVal fnName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- fnGrpName

The string that contains the name of the changed function group.

- fnName

The string that contains the old function information. It can be eVMOperationChange (the name of the old group), eVMOperationAssignmentDelete (the name of the unassigned function), or eVMOperationAssignmentChange (the name of the function that changed in assignment - either changed position within a group or added to the group).

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a new function group is created), eVMOperationDelete (an existing function group is deleted), eVMOperationChange (the function group name or description is changed), or eVMOperationAssignmentDelete (the function is unassigned from the group).

OnMatrixChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in variant/function matrix assignment.

Usage

```
Sub Application_OnMatrixChanged(ByVal varName As String,  
    ByVal fnName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- varName

The string that contains the name of the variant.

- fnName

The string that contains the name of the function.

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationAssignmentChange (function is assigned to variant); eVMOperationAssignmentDelete (function is unassigned from variant).

OnSymbolModificationChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the symbol assignment.

Usage

```
Sub Application_OnSymbolModificationChanged(ByVal symbolName As String,  
    ByVal fnName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- symbolName

The string that contains the name (path) of the changed symbol.

- fnName

The string that contains the name of the function.

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationAssignmentChange (variant and function are assigned); or eVMOperationAssignmentDelete (variant/function assignment is deleted).

OnVariantChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the variant.

Usage

```
Sub Application_OnVariantChanged(ByVal Name As String,  
    ByVal oldName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- Name

The string that contains the name of the changed variant.

- oldName

This argument is valid only for eVMOperationChange. This is a string that contains the old name of the variant (if the variant name changes) or is an empty string (if the description, number, or type field changes.)

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a new variant is created), eVMOperationDelete (an existing variant is deleted), or eVMOperationChange (the variant name, description, number, or type field is changed).

OnVariantGroupChanged Event (Application Object)

Object: [Application Object](#)

Prerequisites: None.

Fires when something has changed in the variant group.

Usage

```
Sub Application_OnVariantGroupChanged(ByVal varGrpName As String,  
    ByVal varName As String,  
    ByVal Operation As EVMEvOperation)
```

Arguments

- varGrpName

The string that contains the name of the changed variant group.

- varName

The string that contains the old variant group information. It can be eVMOperationChange (the name of the old group; if empty, then the changed group description), eVMOperationAssignmentDelete (the name of the variant unassigned from the group), or eVMOperationChange (the name of the variant that changed in assignment - either changed position within a group or added to the group)

- Operation

An enumerate [EVMEvOperation Enum](#). The value can be eVMOperationNew (a new variant group is created), eVMOperationDelete (an existing variant group is deleted), eVMOperationChange (the variant group name or description is changed), eVMOperationAssignmentDelete (the variant is unassigned from the group), eVMOperationAssignmentChange (the variant is assigned to the group).

VMBlockModification Object

The VMBlockModification object represents a modified block. Use this object to access the modified block, its variant and the modification operation. You can also set the operation type for modifying blocks.

The following section defines the methods and properties of the VMBlockModification object.

Table 3. VMBlockModification Object Methods and Properties

Method or Property	Description
BlockSymbol Property (VMBlockModification Object)	Returns the modified block symbol.
ChangeOperation Method (VMBlockModification Object)	Changes the operation for modifying blocks.
Delete Method (VMBlockModification Object)	Deletes the block modification.
Operation Property (VMBlockModification Object)	Returns the type of block modification (replace or unplace).
SubVariantName Property (VMBlockModification Object)	Returns the replaced variant name.
variant Property (VMBlockModification Object)	Returns the modified variant object.

BlockSymbol Property (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified block symbol.

Usage

```
VMBlockModification.BlockSymbol
```

Arguments

None

Return Values

IMGCVMSymbol. The [VMSymbol Object](#).

ChangeOperation Method (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Prerequisites: None.

Changes the operation for modifying blocks.

Usage

```
VMBlockModification.ChangeOperation(ByVal oper As EVMOperation,  
    ByVal subVariant As String) As Integer
```

Arguments

- oper

An enumerate [EVMOperation Enum](#). The value can be eVMOperUnplace (unplace component) or eVMOperReplace (replace block with another block).

Replace is only valid on Reuse Blocks.

- subVariant

A string that contains the variant from the Reuse Block database. Use an empty string for unplace operations.

Return Values

As Integer. Value "1" if operation is successful; value "0" if operation fails.

Examples

```
Block.ChangeOperation(eVMOperUnplace, " ")  
Block.ChangeOperation(eVMOperReplace, "RB Variant_1")
```

Delete Method (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Prerequisites: None.

Deletes the block modification.

Usage

`VMBlockModification.Delete()` As Boolean

Arguments

None

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification is not deleted.

Operation Property (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the type of block modification (replace or unplace).

Usage

VMBlockModification.Operation

Arguments

None

Return Values

EVMOperation. An enumerate ([EVMOperation Enum](#)).

Description

Use the [ChangeOperation](#) on page 49 method to change the operation of a block.

SubVariantName Property (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the replaced variant name.

Usage

```
VMBlockModification.SubVariantName
```

Arguments

None

Return Values

String. A string that contains the replaced variant of block.

Description

Use the [ChangeOperation](#) on page 49 method to change the subVariant name.

variant Property (VMBlockModification Object)

Object: [VMBlockModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified variant object.

Usage

`VMBlockModification.variant`

Arguments

None

Return Values

IMGCVMVariant. The modified variant object ([VMVariant Object](#)).

VMBorderPropModification Object

The VMBorderPropModification object represents a modified border property. Use this object to access the modified border property, its variant and the modification operation.

The following section defines the methods and properties of the VMBorderPropModification object.

Table 4. VMBorderPropModification Object Methods and Properties

Method or Property	Description
BorderSymbol Property (VMBorderPropModification Object)	Returns the modified border symbol.
Delete Method (VMBorderPropModification Object)	Deletes the border property modification.
Name Property (VMBorderPropModification Object)	Returns the name of the border property.
value Property (VMBorderPropModification Object)	Sets or returns the value of the border property.
variant Property (VMBorderPropModification Object)	Returns the modified variant object.

BorderSymbol Property (VMBorderPropModification Object)

Object: [VMBorderPropModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified border symbol.

Usage

`VMBorderPropModification.BorderSymbol`

Arguments

None

Return Values

IMGCVMSymbol. The [VMSymbol Object](#).

Delete Method (VMBorderPropModification Object)

Object: [VMBorderPropModification Object](#)

Prerequisites: None.

Deletes the border property modification.

Usage

```
VMBorderPropModification.Delete() As Boolean
```

Arguments

None

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification is not deleted.

Name Property (VMBorderPropModification Object)

Object: [VMBorderPropModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the name of the border property.

Usage

```
VMBorderPropModification.Name
```

Arguments

None

Return Values

String

value Property (VMBorderPropModification Object)

Object: [VMBorderPropModification Object](#)

Access: Read/Write

Prerequisites: None.

Sets or returns the value of the border property.

Usage

VMBorderPropModification.value

Arguments

None

Return Values

String

variant Property (VMBorderPropModification Object)

Object: [VMBorderPropModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified variant object.

Usage

```
VMBorderPropModification.variant
```

Arguments

None

Return Values

IMGCVMVariant. The modified variant object ([VMVariant Object](#)).

VMBOM Object

The VMBOM object represents the BOM Reports dialog. Use this object to define the format, settings and variants to report in a BOM.

The following section defines the methods and properties of the VMBOM object.

Table 5. VMBOM Object Methods and Properties

Method or Property	Description
Add Method (VMBOM Object)	Adds variant to BOM generation list.
ClearList Method (VMBOM Object)	Removes all variants from the BOM generation list.
Execute Method (VMBOM Object)	Executes BOM generation.
Format Property (VMBOM Object)	Returns or sets the BOM Report format.
GenerateIndex Property (VMBOM Object)	Returns or sets the Generate HTML Master index option.
GenerateReuseBlocksReports Property (VMBOM Object)	Returns or sets the Generate separate Reuse Blocks reports option.
IncludeMaster Property (VMBOM Object)	Returns or sets the Generate Master BOM option.
OutputPath Property (VMBOM Object)	Returns or sets the output path name for BOM reports.
Properties Property (VMBOM Object)	Returns or sets the Properties definition.
Type Property (VMBOM Object)	Returns or sets the Type of BOM report (Part List or Ref Des List).
UserProps Property (VMBOM Object)	Returns or sets the user defined properties.
Variants Property (VMBOM Object)	Returns the collection of variants to generate a BOM.

Add Method (VMBOM Object)

Object: [VMBOM Object](#)

Prerequisites: None.

Adds variant to BOM generation list.

Usage

```
VMBOM.Add(ByVal variant As Variant) As Boolean
```

Arguments

- variant

A string that contains the name of the variant ([VMVariant Object](#)) or the name of the variant group ([VMVariantGroup Object](#)).

Return Values

As Boolean. "True" if at least one correct variant is added; "False" if the variant or variant group is not correct or does not exist.

Description

Adds the name of an individual variant or the name of the variant group (which adds all the variants in the group) to the BOM generation list.

ClearList Method (VMBOM Object)

Object: [VMBOM Object](#)

Prerequisites: None.

Removes all variants from the BOM generation list.

Usage

```
VMBOM.ClearList() As Integer
```

Arguments

None

Return Values

As Integer. Reserved for future use.

Execute Method (VMBOM Object)

Object: [VMBOM Object](#)

Prerequisites: None.

Executes BOM generation.

Usage

VMBOM.Execute() As Integer

Arguments

None

Return Values

As Integer. Reserved for future use.

Format Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the BOM Report format.

Usage

```
VMBOM.Format = EVMReportFormat
```

Arguments

None

Return Values

EVMReportFormat. An enumerate [EVMReportFormat Enum](#).

GenerateIndex Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Generate HTML Master index option.

Usage

VMBOM.GenerateIndex = True | False

Arguments

None

Return Values

As Boolean. "True" to generate an index file; "False" to not generate the index.

GenerateReuseBlocksReports Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Generate separate Reuse Blocks reports option.

Usage

VMBOM.GenerateReuseBlocksReports = True | False

Arguments

None

Return Values

As Boolean. "True" to generate separate reports for each reuse block; "False" to generate a single report for all reuse blocks.

IncludeMaster Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Generate Master BOM option.

Usage

VMBOM.IncludeMaster = True | False

Arguments

None

Return Values

As Boolean. "True" to generate the master BOM; "False" to not generate the master BOM.

OutputPath Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the output path name for BOM reports.

Usage

```
VMBOM.OutputPath = String
```

Arguments

None

Return Values

String. The output path where BOM reports are generated.

Description

The path contains the directory and prefix file name where the BOM reports are generated. The generated names are created using the schematic OutputPath + “_” + variant name (or “_master”) and the extension based on the Format Property. For example:

Generated master files have the following output file format.

```
“C:\Reports\Output\out__master.xxx”
```

Generated variant files have the following output file format.

```
“C:\Reports\Output\out_Variant1.xxx”
```


Properties Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Properties definition.

Usage

```
VMBOM.Properties = String
```

Arguments

None

Return Values

String. The properties from the LibDataGrid connection. For example:
“\$(DEVICE)\$(PKG_TYPE)\$(TOLERANCE)\$(VALUE)\$(SYM)\$(Library)”

Description

Defines the properties to add from the databases based on the format information in the Library Query Setup tab sheet. For format information, refer to “Defining the library data source” section of the *Variant Manager User’s Manual*.

Type Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Type of BOM report (Part List or Ref Des List).

Usage

```
VMBOM.Type = EVMBOMType
```

Arguments

None

Return Values

EVMBOMType. An enumerate [EVMBOMType Enum](#). The value can be either a eVMBOMPartList (part list) or eVMBOMRefDesList (reference designator list).

Description

The BOM report generates as either a part list or a reference designator list.

UserProps Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the user defined properties.

Usage

VMBOM.UserProps As String

Arguments

None

Return Values

String. The user defined properties.

Variants Property (VMBOM Object)

Object: [VMBOM Object](#)

Access: Read-Only

Prerequisites: None.

Returns the collection of variants to generate a BOM.

Usage

VMBOM.Variants

Arguments

None

Return Values

IMGCVMVariants. The collection of variants [VMVariants Collection](#). If empty, the collection size is "0".

VMColor Object

The VMColor object represent the red, green, blue values for a color object.

The following section defines the properties of the VMColor object.

Table 6. VMColor Object Properties

Property	Description
blue Property (VMColor Object)	Returns or sets the Blue color value.
green Property (VMColor Object)	Returns or sets the Green color value.
red Property (VMColor Object)	Returns or sets the Red color value.

blue Property (VMColor Object)

Object: [VMColor Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Blue color value.

Usage

```
VMColor.blue = Long
```

Arguments

None

Return Values

Long. The blue value (0-255).

green Property (VMColor Object)

Object: [VMColor Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Green color value.

Usage

```
VMColor.green = Long
```

Arguments

None

Return Values

Long. The green value (0-255).

red Property (VMColor Object)

Object: [VMColor Object](#)

Access: Read/Write

Prerequisites: None.

Returns or sets the Red color value.

Usage

```
VMColor.red = Long
```

Arguments

None

Return Values

Long. The red value (0-255).

VMComponent Object

The VMComponent object represents a physical component. Use the object to access attributes of the component including its name, current and original part numbers, the symbols that define the component and the variant associated with the component.

The following section defines the methods and properties of the VMComponent object.

Table 7. VMComponent Object Properties

Property	Description
AliasPartNumber Property (VMComponent Object)	Returns the current alias part number used by the component.
ContainsNested Property (VMComponent Object)	Returns True if component has nested components.
Name Property (VMComponent Object)	Returns the component Reference Designator, cell name (mechanical) or master component name (nested).
OriginalAliasPartNumber Property (VMComponent Object)	Returns the master alias part number used by the component.
OriginalPartNumber Property (VMComponent Object)	Returns the part number of the master component.
ParentComponent Property (VMComponent Object)	Returns parent component (for nested components only).
PartNumber Property (VMComponent Object)	Returns the current part number of the component.
SID Property (VMComponent Object)	Returns the unique id of mechanical cells.
Symbols Property (VMComponent Object)	Returns the collection of symbols that define the component.
Type Property (VMComponent Object)	Returns the component type.
variant Property (VMComponent Object)	Returns the variant object associated with the component.

AliasPartNumber Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the current alias part number used by the component.

Usage

```
VMComponent.AliasPartNumber
```

Arguments

None

Return Values

String. A string that contains the alias part number used by the component.

Description

OriginalAliasPartNumber and AliasPartNumber properties are the same for unmodified components.

If you replace the variant component object, then the alias part number in the variant returns.

ContainsNested Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns True if component has nested components.

Usage

`VMComponent.ContainsNested`

Arguments

None

Return Values

Boolean. If True, the component has nested components. If False, the component has no nested components.

Name Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the component Reference Designator, cell name (mechanical) or master component name (nested).

Usage

VMComponent.Name

Arguments

None

Return Values

String

- regular component, contains the reference designator
- mechanical part, contains the cell name
- nested logical component. contains the master component name (for example, U1/Part_001")

OriginalAliasPartNumber Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the master alias part number used by the component.

Usage

VMComponent.OriginalAliasPartNumber

Arguments

None

Return Values

String. A string that contains the master alias part number used by the component.

Description

OriginalAliasPartNumber and AliasPartNumber properties are the same for unmodified components.

OriginalPartNumber Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the part number of the master component.

Usage

```
VMComponent.OriginalPartNumber
```

Arguments

None

Return Values

String. A string that contains the part number of the master component.

Description

OriginalPartNumber and PartNumber properties are the same for unmodified components.

ParentComponent Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns parent component (for nested components only).

Usage

`VMComponent.ParentComponent`

Arguments

None

Return Values

IMGCVMComponent. The parent [VMComponent Object](#).

PartNumber Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the current part number of the component.

Usage

`VMComponent.PartNumber`

Arguments

None

Return Values

String. A string that contains the part number of the component.

Description

OriginalPartNumber and PartNumber properties are the same for unmodified components.

If you replace the variant component object, then the part number in the variant returns.

SID Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the unique id of mechanical cells.

Usage

VMComponent.SID

Arguments

None

Return Values

String. A string that contains the mechanical cell's unique id.

Symbols Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the collection of symbols that define the component.

Usage

`VMComponent.Symbols`

Arguments

None

Return Values

IMGCVMSymbols. The collection of symbols ([VMSymbols Collection](#)) that define the component.

Type Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the component type.

Usage

VMComponent.Type

Arguments

None

Return Values

EVMComponentType. The component type ([EVMComponentType Enum](#)).

variant Property (VMComponent Object)

Object: [VMComponent Object](#)

Access: Read-Only

Prerequisites: None.

Returns the variant object associated with the component.

Usage

VMComponent.variant

Arguments

None

Return Values

IMGCVMVariant. The [VMVariant Object](#). Returns Nothing if there is no variant associated with the component.

VMComponentModification Object

The VMComponentModification object represents a modified component. Use this object to access the modified component, its variant and the modification operation. You can also set the operation type for modifying components.

The following section defines the methods and properties of the VMComponentModification object.

Table 8. VMComponentModification Object Methods and Properties

Method or Property	Description
ChangeOperation Method (VMComponentModification Object)	Changes the operation for modifying components.
ChangeOperationEx Method (VMComponentModification Object)	Changes the component modification with the alias part number.
Component Property (VMComponentModification Object)	Returns the modified component object.
Delete Method (VMComponentModification Object)	Deletes the component modification.
newAliasPartNumber Property (VMComponentModification Object)	Returns the new alias part number of the component (for replace modification).
NewPartNumber Property (VMComponentModification Object)	Returns the new part number of the component (for replace modification).
Operation Property (VMComponentModification Object)	Returns the type of component modification (replace or unplace).
variant Property (VMComponentModification Object)	Returns the modified variant object.

ChangeOperation Method (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Prerequisites: None.

Changes the operation for modifying components.

Usage

```
VMComponentModification.ChangeOperation(ByVal oper As EVMOperation,  
    ByVal NewPartNumber As String,  
    [ByVal cellName As String]) As Integer
```

Arguments

- oper

An enumerate [EVMOperation Enum](#). The value can be eVMOperUnplace (unplace component) or eVMOperReplace (replace component with another part).

- NewPartNumber

A string that contains the new part number (for replaced components).

- cellName

(Optional) Reserved for future use.

Return Values

As Integer. Value “1” if operation is successful; value “0” if operation fails.

ChangeOperationEx Method (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Prerequisites: None.

Changes the component modification with the alias part number.

Usage

```
VMComponentModification.ChangeOperationEx(ByVal oper As EVMOperation,  
    ByVal NewPartNumber As String,  
    ByVal newAliasPartNumber As String,  
    [ByVal cellName As String]) As Integer
```

Arguments

- oper

An enumerate [EVMOperation Enum](#). The value can be eVMOperUnplace (unplace component) or eVMOperReplace (replace component with another part).

- NewPartNumber

A string that contains the new part number (for replaced components).

- newAliasPartNumber

(Optional) A string that contains the new alias part number (for replaced components).

- cellName

(Optional) Reserved for future use.

Return Values

As Integer. Value “1” if operation is successful; value “0” if operation fails.

Component Property (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified component object.

Usage

`VMComponentModification.Component`

Arguments

None

Return Values

IMGCVMComponent. The [VMComponent Object](#).

Delete Method (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Prerequisites: None.

Deletes the component modification.

Usage

```
VMComponentModification.Delete() As Boolean
```

Arguments

None

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification is not deleted.

newAliasPartNumber Property (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the new alias part number of the component (for replace modification).

Usage

```
VMComponentModification.newAliasPartNumber
```

Arguments

None

Return Values

String. A string that contains the new alias part number of the component.

Description

Use the [“ChangeOperationEx”](#) on page 91 method to change the alias part number.

NewPartNumber Property (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the new part number of the component (for replace modification).

Usage

`VMComponentModification.NewPartNumber`

Arguments

None

Return Values

String. A string that contains the new part number of component.

Description

Use the [ChangeOperation](#) on page 90 method to change the part number.

Operation Property (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the type of component modification (replace or unplace).

Usage

`VMComponentModification.Operation`

Arguments

None

Return Values

EVMOperation. An enumerate ([EVMOperation Enum](#)).

Description

Use the [ChangeOperation](#) on page 90 method to change type of component modification.

variant Property (VMComponentModification Object)

Object: [VMComponentModification Object](#)

Access: Read-Only

Prerequisites: None.

Returns the modified variant object.

Usage

`VMComponentModification.variant`

Arguments

None

Return Values

IMGCVMVariant. The modified variant object ([VMVariant Object](#)).

VMDocument Object

The VMDocument object represent a Variant Manager design. Use this object to delete, find and add Variant Manager design objects (Components, Symbols, Functions, Variants, Function Groups, Variant Groups, Component Modifications and Symbol Modifications).

The following section defines the methods and properties of the VMDocument object.

Table 9. VMDocument Object Methods and Properties

Method or Property	Description
AssignFunctionToVariant Method (VMDocument Object)	Assigns a function to a FMV variant.
BlockModifications Property (VMDocument Object)	Returns the collection of variant block modifications.
BorderPropModifications Property (VMDocument Object)	Returns the collection of variant border property modifications.
ComponentModifications Property (VMDocument Object)	Returns the collection of variant component modifications.
Components Property (VMDocument Object)	Returns the collection of variant components.
DeleteBlockModification Method (VMDocument Object)	Deletes block modification.
DeleteBorderPropModification Method (VMDocument Object)	Deletes border property modification.
DeleteComponentModification Method (VMDocument Object)	Deletes component modifications.
DeleteFunction Method (VMDocument Object)	Deletes function object.
DeleteFunctionGroup Method (VMDocument Object)	Deletes a function group object.
DeleteSymbolModification Method (VMDocument Object)	Deletes the symbol modification object.
DeleteVariant Method (VMDocument Object)	Deletes the variant object.
DeleteVariantGroup Method (VMDocument Object)	Deletes the variant group object.
FindBlockModification Method (VMDocument Object)	Returns the block modification object.
FindBorderPropModification Method (VMDocument Object)	Returns the border property modification.
FindComponent Method (VMDocument Object)	Returns the component object.
FindComponentModification Method (VMDocument Object)	Returns the component modification object.
FindFunction Method (VMDocument Object)	Returns the function object.
FindFunctionGroup Method (VMDocument Object)	Returns the function group object.

Table 9. VMDocument Object Methods and Properties (continued)

Method or Property	Description
FindSymbol Method (VMDocument Object)	Returns the symbol object.
FindSymbolModification Method (VMDocument Object)	Returns the symbol modification object.
FindVariant Method (VMDocument Object)	Returns a variant object.
FindVariantGroup Method (VMDocument Object)	Returns a variant group object.
FunctionGroups Property (VMDocument Object)	Returns the collection of defined function groups.
Functions Property (VMDocument Object)	Returns the collection of defined functions.
GetBorderSymbols Method (VMDocument Object)	Returns the collection of border symbols.
LockRedraw Method (VMDocument Object)	Locks the graphical user interface to prevent the design from updating.
PutBlockModification Method (VMDocument Object)	Creates a block modification object.
PutBorderPropModification Method (VMDocument Object)	Creates a modification on a border property.
PutComponentModification Method (VMDocument Object)	Creates a component modification object.
PutFunction Method (VMDocument Object)	Creates a new function object.
PutFunctionGroup Method (VMDocument Object)	Creates a new function group.
PutSymbolModification Method (VMDocument Object)	Creates a symbol modification object.
PutVariant Method (VMDocument Object)	Creates a new variant.
PutVariantGroup Method (VMDocument Object)	Creates a new variant group.
SelectVariantGroup Method (VMDocument Object)	Selects the variant group.
SymbolModifications Property (VMDocument Object)	Returns the collection of symbol modifications.
Symbols Property (VMDocument Object)	Returns the collection of symbols.
UnAssignFunctionToVariant Method (VMDocument Object)	Unassigns a function from a variant.
UnLockRedraw Method (VMDocument Object)	Unlocks the graphical user interface.
VariantGroups Property (VMDocument Object)	Returns the collection of variant groups.
Variants Property (VMDocument Object)	Returns the collection of variants.

AssignFunctionToVariant Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Assigns a function to a FMV variant.

Usage

```
VMDocument.AssignFunctionToVariant(ByVal function As Variant,  
    ByVal variant As Variant) As Integer
```

Arguments

- function

This can be a Function name or [VMFunction Object](#).

- variant

This can be a FMV Variant name or [VMVariant Object](#).

Return Values

As Integer. Value “1” if assignment changes; value “0” if assignment fails.

BlockModifications Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variant block modifications.

Usage

```
VMDocument.BlockModifications([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The variant name or [VMVariant Object](#) to filter on. If not specified, the property returns all block modifications.

Return Values

IMGCVMBlockModifications. The collection of block modifications ([VMBlockModifications Collection](#)) for the variant.

BorderPropModifications Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variant border property modifications.

Usage

```
VMDocument.BorderPropModifications([ByVal filter As Variant]) IMGCVMBorderPropModifications
```

Arguments

- filter

(Optional) The variant name or [VMVariant Object](#) to filter on. If not specified, the property returns all border property modifications.

Return Values

IMGCVMBorderPropModifications. The collection of border property modifications ([VMBlockModifications Collection](#)) for the variant.

ComponentModifications Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variant component modifications.

Usage

`VMDocument.ComponentModifications([ByVal filter As Variant])`

Arguments

- filter

(Optional) The variant name or [VMVariant Object](#) to filter on. If not specified, the property returns all component modifications.

Return Values

IMGCVMComponentModifications. The collection of component modifications ([VMComponentModifications Collection](#)) for the variant.

Components Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variant components.

Usage

```
VMDocument.Components([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The variant name or [VMVariant Object](#) to filter on. If not specified, the property returns all placed components.

Return Values

IMGCVMComponents. The collection of components: ([VMComponents Collection](#)) for the variant. Unplaced components are not included.

DeleteBlockModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes block modification.

Usage

```
VMDocument.DeleteBlockModification(ByVal variantDefinition As Variant,  
    ByVal blockDefinition As Variant) As Boolean
```

Arguments

- variantDefinition

This can be a Variant name or [VMVariant Object](#).

- blockDefinition

This can be a Symbol name (full path) or [VMSymbol Object](#).

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification parameters are wrong or do not exist.

DeleteBorderPropModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes border property modification.

Usage

```
VMDocument.DeleteBorderPropModification(ByVal variantDefinition As Variant, ByVal  
borderSymbolDefinition As Variant, ByVal propName As String) As Boolean
```

Arguments

- variantDefinition

This can be a Variant name or [VMVariant Object](#).

- borderSymbolDefinition

This can be a Symbol name (full path) or [VMSymbol Object](#).

- propName

This is a string that corresponds to the border modification property to delete.

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification parameters are wrong or do not exist.

DeleteComponentModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes component modifications.

Usage

```
VMDocument.DeleteComponentModification(ByVal variantDefinition As Variant,  
    ByVal componentDefinition As Variant) As Boolean
```

Arguments

- variantDefinition

This can be a Variant name or [VMVariant Object](#).

- componentDefinition

This can be a Component name or [VMComponent Object](#).

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification parameters are wrong or do not exist.

DeleteFunction Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes function object.

Usage

```
VMDocument.DeleteFunction(ByVal functionToDelete As Variant)
```

Arguments

- functionToDelete

This can be a Function name or [VMFunction Object](#).

DeleteFunctionGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes a function group object.

Usage

VMDocument.DeleteFunctionGroup(ByVal toDelete As Variant)

Arguments

- toDelete

This can be a Function Group name or [VMFunctionGroup Object](#).

DeleteSymbolModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes the symbol modification object.

Usage

```
VMDocument.DeleteSymbolModification(ByVal functionDefinition As Variant,  
    ByVal symbolDefinition As Variant) As Boolean
```

Arguments

- functionDefinition

This can be a Function name or [VMFunction Object](#).

- symbolDefinition

This can be a Symbol name (full path) or [VMSymbol Object](#).

Return Values

As Boolean. "True" if the modification is deleted; "False" if the modification parameters are wrong or do not exist.

DeleteVariant Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes the variant object.

Usage

```
VMDocument.DeleteVariant(ByVal variantToDelete As Variant)
```

Arguments

- variantToDelete

This can be a Variant name or [VMVariant Object](#).

DeleteVariantGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Deletes the variant group object.

Usage

```
VMDocument.DeleteVariantGroup(ByVal toDelete As Variant)
```

Arguments

- toDelete

This can be a Variant Group name or [VMVariantGroup Object](#).

FindBlockModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the block modification object.

Usage

```
VMDocument.FindBlockModification(  
    ByVal variantDefinition As Variant,  
    ByVal blockDefinition As Variant) As IMGCVMBlockModificationt
```

Arguments

- variantDefinition

This can be a Variant name or [VMVariant Object](#).

- blockDefinition

This can be a Symbol name (full path) or [VMSymbol Object](#).

Return Values

As IMGCVMBlockModificationt. The [VMBlockModification Object](#). Returns Nothing if the modification cannot be found.

FindBorderPropModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the border property modification.

Usage

```
VMDocument.FindBorderPropModification(  
    ByVal variantDefinition As Variant,  
    ByVal SymbolDefinition As Variant,  
    ByVal propName As String) As IMGCVMBorderPropModification
```

Arguments

- variantDefinition
This can be a Variant name or [VMVariant Object](#).
- SymbolDefinition
This can be a Symbol name (full path) or [VMSymbol Object](#).
- propName
A string that identifies the border property.

Return Values

As IMGCVMBorderPropModification. The [VMBorderPropModification Object](#). Return Nothing if the modification cannot be found.

FindComponent Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the component object.

Usage

```
VMDocument.FindComponent(ByVal componentName As String) As IMGCVMSComponent
```

Arguments

- componentName

A string that contains the name of the component (reference designator).

Return Values

As IMGCVMSComponent. The [VMComponent Object](#). Returns Nothing if the component cannot be found.

FindComponentModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the component modification object.

Usage

```
VMDocument.FindComponentModification(ByVal variantDefinition As Variant,  
    ByVal componentDefinition As Variant) As IMGCVMSComponentModification
```

Arguments

- variantDefinition

This can be a Variant name or [VMVariant Object](#).

- componentDefinition

This can be a Component name or [VMComponent Object](#).

Return Values

As IMGCVMSComponentModification. The [VMComponentModification Object](#). Returns Nothing if the modification cannot be found.

FindFunction Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the function object.

Usage

```
VMDocument.FindFunction(ByVal Name As String) As IMGCVFunction
```

Arguments

- Name

A string that contains the function name.

Return Values

As IMGCVFunction. The [VMFunction Object](#). Returns Nothing if the function cannot be found.

FindFunctionGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the function group object.

Usage

```
VMDocument.FindFunctionGroup(ByVal GroupName As String) As IMGCVMLFunctionGroup
```

Arguments

- GroupName

A string that contains the function group name.

Return Values

As IMGCVMLFunctionGroup. The [VMFunctionGroup Object](#). Returns Nothing if the function group does not exist.

FindSymbol Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the symbol object.

Usage

```
VMDocument.FindSymbol(ByVal symbolNameAndPath As String) As IMGCVMSymbol
```

Arguments

- symbolNameAndPath

A string that contains the symbol identification (full path including name).

Return Values

As IMGCVMSymbol. The [VMSymbol Object](#). Returns Nothing if the symbol is not found.

FindSymbolModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns the symbol modification object.

Usage

```
VMDocument.FindSymbolModification(ByVal functionDefinition As Variant,  
    ByVal symbolDefinition As Variant) As IMGCVMSymbolModification
```

Arguments

- functionDefinition

This can be a Function name or [VMFunction Object](#).

- symbolDefinition

This can be a Symbol name (full path) or [VMSymbol Object](#).

Return Values

As IMGCVMSymbolModification. The [VMSymbolModification Object](#). Returns Nothing if the function and the symbol modification do not exist.

FindVariant Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns a variant object.

Usage

```
VMDocument.FindVariant(ByVal Name As String) As IMGCVMMVariant
```

Arguments

- Name

A string that contains the variant name.

Return Values

As IMGCVMMVariant. The [VMVariant Object](#). Returns Nothing if the variant is not found.

FindVariantGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Returns a variant group object.

Usage

```
VMDocument.FindVariantGroup(ByVal GroupName As String) As IMGCVMSVariantGroup
```

Arguments

- GroupName

A string that contains the variant group name.

Return Values

As IMGCVMSVariantGroup. The [VMVariantGroup Object](#). Returns Nothing if the variant group is not found.

FunctionGroups Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of defined function groups.

Usage

```
VMDocument.FunctionGroups([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The function name or [VMFunction Object](#) to filter on. If not specified, the property returns all function groups.

Return Values

IMGCVMFunctionGroups. The collection of all function groups ([VMFunctionGroups Collection](#)) that have the function assigned.

Functions Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of defined functions.

Usage

VMDocument.Functions([ByVal filter As Variant])

Arguments

- filter

(Optional) A function group name, [VMFunctionGroup Object](#), variant name or [VMVariant Object](#). If not specified, the property returns all functions.

Return Values

IMGCVMFunctions. The collection of defined functions ([VMFunctions Collection](#)):

- With a function group filter, the property returns the collection of functions in the function group
- With a variant filter, the property returns the collection of functions assigned the variant

GetBorderSymbols Method (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of border symbols.

Usage

```
VMDocument.GetBorderSymbols() As IMGCVMSymbols
```

Arguments

None

Return Values

As IMGCVMSymbols. The [VMSymbols Collection](#). Returns Nothing if no symbols are found.

LockRedraw Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Locks the graphical user interface to prevent the design from updating.

Usage

```
VMDocument.LockRedraw() As Long
```

Arguments

None

Return Values

As Long. The identification of the lock. Use the identifier with the UnLockRedraw method.

PutBlockModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a block modification object.

Usage

```
VMDocument.PutBlockModification(ByVal variantDefinition As Variant,  
    ByVal blockDefinition As Variant,  
    ByVal changeOper As EVMOperation,  
    [ByVal subVariantName As String]) As IMGCVMBlockModification
```

Arguments

- variantDefinition
The variant name or [VMVariant Object](#).
- blockDefinition
The symbol name (full path) or [VMSymbol Object](#).
- changeOper
The operation type ([EVMOperation Enum](#))- eVMOperUnplace to unplace a block;
eVMOperReplace to replace a block
- subVariantName
(Optional) Valid for replacing on reuse block. A string that contains the variant name from the
reuse block database.

Return Values

As IMGCVMBlockModification. The newly created [VMBlockModification Object](#).

PutBorderPropModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a modification on a border property.

Usage

```
VMDocument.PutBorderPropModification(ByVal variantDefinition As Variant,  
    ByVal borderSymbolDefinition As Variant,  
    ByVal propName As String,  
    ByVal propValue As String) As IMGCVMBorderPropModification
```

Arguments

- variantDefinition
The variant definition or [VMVariant Object](#).
- borderSymbolDefinition
The variant border symbol or [VMSymbol Object](#).
- propName
The name of the property to be modified.
- propValue
The value for the border property.

Return Values

As IMGCVMBorderPropModification. The newly created [VMBorderPropModification Object](#).

PutComponentModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a component modification object.

Usage

```
VMDocument.PutComponentModification(ByVal variantDefinition As Variant,  
    ByVal componentDefinition As Variant,  
    ByVal changeOper As EVMOperation,  
    [ByVal PartNumber As String],  
    [ByVal cellName As String]) As IMGCVMComponentModification
```

Arguments

- variantDefinition
The variant name or [VMVariant Object](#).
- componentDefinition
The component name or [VMComponent Object](#).
- changeOper
The operation type ([EVMOperation Enum](#))- eVMOperUnplace to unplace a component;
eVMOperReplace to replace a component
- PartNumber
(Optional) A string that contains the new part number (replace mode).
- cellName
(Optional) Reserved for future use.

Return Values

As IMGCVMComponentModification. The newly created [VMComponentModification Object](#).

PutFunction Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a new function object.

Usage

```
VMDocument.PutFunction(ByVal FunctionName As String,  
    ByVal FunctionName As String,  
    ByVal FunctionNumber As String,  
    ByVal FunctionDescription As String,  
    [ByVal sourceForContents As Variant]) As IMGCVFunction
```

Arguments

- FunctionName

A string that contains the name of the new function.

- FunctionNumber

A string that contains the number of the new function.

- FunctionDescription

A string that contains the description of the new function.

- sourceForContents

(Optional) A function name or [VMFunction Object](#). This is a template.

Return Values

As IMGCVFunction. The newly created [VMFunction Object](#).

PutFunctionGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a new function group.

Usage

```
VMDocument.PutFunctionGroup(ByVal GroupName As String,  
    ByVal GroupDescription As String,  
    [ByVal sourceForContents As Variant]) As IMGCVMFunctionGroup
```

Arguments

- GroupName

A string that contains the name of new function group.

- GroupDescription

A string that contains the description of new function group.

- sourceForContents

(Optional) A function group name or [VMFunctionGroup Object](#). This is a template.

Return Values

As IMGCVMFunctionGroup. The newly created [VMFunctionGroup Object](#).

PutSymbolModification Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a symbol modification object.

Usage

```
VMDocument.PutSymbolModification(ByVal functionDefinition As Variant,  
    ByVal symbolDefinition As Variant,  
    ByVal changeOper As EVMFMVOperation,  
    [ByVal PartNumber As String],  
    [ByVal cellName As String]) As IMGCVMSymbolModification
```

Arguments

- functionDefinition
The function name or [VMFunction Object](#).
- symbolDefinition
The symbol name (full path) or [VMSymbol Object](#).
- changeOper
The type of modification ([EVMFMVOperation Enum](#)).
- PartNumber
(Optional) A string that contains the new part number (replace modifications).
- cellName
(Optional) Reserved for future use.

Return Values

As IMGCVMSymbolModification. The newly created [VMSymbolModification Object](#).

PutVariant Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a new variant.

Usage

```
VMDocument.PutVariant(ByVal variantName As String,  
    ByVal variantNumber As String,  
    ByVal VariantDescription As String,  
    [ByVal sourceForContents As Variant]) As IMGCVMVariant
```

Arguments

- variantName
A string that contains the name of the new variant.
- variantNumber
A string that contains the number of the new variant.
- variantDescription
A string that contains the description of the new variant.
- sourceForContents
(Optional) A variant name or [VMVariant Object](#). This is a template.

Return Values

As IMGCVMVariant. The newly created [VMVariant Object](#).

PutVariantGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Creates a new variant group.

Usage

```
VMDocument.PutVariantGroup(ByVal GroupName As String,  
    ByVal GroupDescription As String,  
    [ByVal sourceForContents As Variant]) As IMGCVMVariantGroup
```

Arguments

- GroupName

A string that contains the new variant group name.

- GroupDescription

A string that contains the description of the variant group.

- sourceForContents

(Optional) A variant group name or [VMVariantGroup Object](#). This is a template.

Return Values

As IMGCVMVariantGroup. The newly created [VMVariantGroup Object](#).

SelectVariantGroup Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Selects the variant group.

Usage

```
VMDocument.SelectVariantGroup(ByVal GroupName As String) As Boolean
```

Arguments

- GroupName

A string that contains the variant group name.

Return Values

Boolean. If True, select the variant group. If False, unselect the variant group.

SymbolModifications Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of symbol modifications.

Usage

`VMDocument.SymbolModifications([ByVal filter As Variant])`

Arguments

- filter

(Optional) The function name, [VMFunction Object](#), symbol name (full path) or [VMSymbol Object](#). If not specified, the property returns all symbol modifications.

Return Values

IMGCVMSymbolModifications. The collection of symbol modifications ([VMSymbolModifications Collection](#))

- With function filter, the property returns the collection of all modifications in the function.
- With symbol filter, the property returns the collection of all modifications on the symbol.

Symbols Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of symbols.

Usage

`VMDocument.Symbols([ByVal filter As Variant])`

Arguments

- filter

(Optional) The function name, or [VMFunction Object](#). If not specified, the property returns all symbols on the top level of hierarchy.

Return Values

IMGCVMSymbols. The [VMSymbols Collection](#) of symbols assigned the function.

UnAssignFunctionToVariant Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Unassigns a function from a variant.

Usage

```
VMDocument.UnAssignFunctionToVariant(ByVal function As Variant,  
    ByVal variant As Variant) As Integer
```

Arguments

- function
The function name or [VMFunction Object](#).
- variant
The variant name or [VMVariant Object](#).

Return Values

As Integer. Value "1" if successful. Value "0" if unsuccessful.

UnLockRedraw Method (VMDocument Object)

Object: [VMDocument Object](#)

Prerequisites: None

Unlocks the graphical user interface.

Usage

```
VMDocument.UnLockRedraw(lockID As Long,  
    [forFutureUse As Boolean = True]) As Boolean
```

Arguments

- lockID
Use the identifier returned by the [LockRedraw](#) on page 126 method.
- forFutureUse
Reserved for future use.

Return Values

As Boolean. If True, the unlock was successful. If False, the unlock failed.

VariantGroups Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variant groups.

Usage

```
VMDocument.VariantGroups([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The variant name or [VMVariant Object](#). If not specified, the property returns all variant groups.

Return Values

IMGCVariantGroups. The collection of defined variant groups ([VMVariantGroups Collection](#)) that include the variant.

Variants Property (VMDocument Object)

Object: [VMDocument Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variants.

Usage

```
VMDocument.Variants([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The variant group name, [VMVariantGroup Object](#), function name or [VMFunction Object](#). If not specified, the property returns all variants.

Return Values

IMGCVMMVariants. The [VMVariants Collection](#):

- With variant group filter, the property returns the collection of variants in the group.
- With function filter, the property returns the collection of variants assigned the function.

VMFunctionGroup Object

The VMFunctionGroup object represents a function group. Use this object to assign and unassign functions to a function group and to access the name, description and functions of the function group.

The following section defines the methods and properties of the VMFunctionGroup object.

Table 10. VMFunctionGroup Object Methods and Properties

Method or Property	Description
Assign Method (VMFunctionGroup Object)	Assigns a function to the list of assigned functions.
AssignAll Method (VMFunctionGroup Object)	Assigns all functions to the list of assigned functions.
Description Property (VMFunctionGroup Object)	Returns or sets the description of the function group.
Functions Property (VMFunctionGroup Object)	Returns the collection of functions assigned to the function group.
Name Property (VMFunctionGroup Object)	Returns or sets the name of the function group
UnAssign Method (VMFunctionGroup Object)	Unassigns a function from the list of assigned functions.
UnAssignAll Method (VMFunctionGroup Object)	Unassigns all functions from the list of assigned functions.

Assign Method (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Prerequisites: None

Assigns a function to the list of assigned functions.

Usage

```
VMFunctionGroup.Assign(ByVal nameToAssign As Variant,  
    [ByVal position As Integer = -1]) As Integer
```

Arguments

- nameToAssign

The function name or [VMFunction Object](#)

- position

(Optional) The position of the added function. Specify -1 to add to the end of the list.

Return Values

As Integer. Reserved for future use.

AssignAll Method (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Prerequisites: None

Assigns all functions to the list of assigned functions.

Usage

```
VMFunctionGroup.AssignAll() As Integer
```

Arguments

None

Return Values

As Integer. Reserved for future use.

Description Property (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the description of the function group.

Usage

```
VMFunctionGroup.Description = String
```

Arguments

None

Return Values

String. The description of the group.

Functions Property (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of functions assigned to the function group.

Usage

```
VMFunctionGroup.Functions
```

Arguments

None

Return Values

IMGCVMFunctions. The collection of assigned functions ([VMFunctions Collection](#)).

Name Property (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the name of the function group

Usage

```
VMFunctionGroup.Name = String
```

Arguments

None

Return Values

String. The name of the function group.

UnAssign Method (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Prerequisites: None

Unassigns a function from the list of assigned functions.

Usage

```
VMFunctionGroup.UnAssign(ByVal nameToAssign As Variant) As Integer
```

Arguments

- nameToAssign

The function name or [VMFunction Object](#).

Return Values

As Integer. Value “1” if successful; value “0” if the function is incorrect or is not assigned to the function group.

UnAssignAll Method (VMFunctionGroup Object)

Object: [VMFunctionGroup Object](#)

Prerequisites: None

Unassigns all functions from the list of assigned functions.

Usage

```
VMFunctionGroup.UnAssignAll() As Integer
```

Arguments

None

Return Values

As Integer. Value "1" if successful; value "0" if the default group cannot be modified.

VMFunction Object

The VMFunction object represents a function. Use this object to access the name and description of a function.

The following section defines the properties of the VMFunction object.

Table 11. VMFunction Object Properties

Property	Description
Description Property (VMFunction Object)	Returns or sets the description of the function.
Name Property (VMFunction Object)	Returns or sets the name of the function.

Description Property (VMFunction Object)

Object: [VMFunction Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the description of the function.

Usage

```
VMFunction.Description = String
```

Arguments

None

Return Values

String. The description of the function.

Name Property (VMFunction Object)

Object: [VMFunction Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the name of the function.

Usage

```
VMFunction.Name = String
```

Arguments

None

Return Values

String. The name of the function.

VMFXEDData Object

The VMFXEDData object represent the Xpedition FabLink data. Use the object to add (or clear) variants or variant groups to the assigned list, or to access the variants currently assigned to the list. This object also performs the data generation.

The following section defines the methods and properties of the VMFXEDData object.

Table 12. VMFXEDData Object Methods and Properties

Method or Property	Description
Add Method (VMFXEDData Object)	Adds a variant or variant group to the list for FXE Data generation.
ClearList Method (VMFXEDData Object)	Clears the list of variants for FXE data generation.
Execute Method (VMFXEDData Object)	Executes FXE data generation.
Variants Property (VMFXEDData Object)	Returns the collection of variants assigned to the FXE Generation list.

Add Method (VMFXEDData Object)

Object: [VMFXEDData Object](#)

Prerequisites: None

Adds a variant or variant group to the list for FXE Data generation.

Usage

```
VMFXEDData.Add(ByVal variant As Variant) As Boolean
```

Arguments

- variant

A variant name, [VMVariant Object](#), variant group name, [VMVariantGroup Object](#) or “All”.

Return Values

As Boolean. “True” if successful; “False” if the variant or variant group cannot be added.

ClearList Method (VMFXEDData Object)

Object: [VMFXEDData Object](#)

Prerequisites: None

Clears the list of variants for FXE data generation.

Usage

```
VMFXEDData.ClearList() As Integer
```

Arguments

None

Return Values

As Integer. Value "1" always.

Execute Method (VMFXEDData Object)

Object: [VMFXEDData Object](#)

Prerequisites: None

Executes FXE data generation.

Usage

VMFXEDData.Execute() As Integer

Arguments

None

Return Values

As Integer. Value "1" if successful; value "0" if not executed.

Variants Property (VMFXEDData Object)

Object: [VMFXEDData Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variants assigned to the FXE Generation list.

Usage

VMFXEDData.Variants

Arguments

None

Return Values

IMGCVMVariants. The [VMVariants Collection](#).

VMGUIProperty Object

The VMGUIProperty object represents a name/value property pair. Use this object to access properties. Refer to the VMVariant object for methods and properties that add, delete and find GUIProperty objects. The following section defines the methods and properties of the VMGUIProperty object.

Table 13. VMGUIProperty Object Methods and Properties

Method or Property	Description
Delete Method (VMGUIProperty Object)	Deletes the property.
Name Property (VMGUIProperty Object)	Returns the name of the property.
value Property (VMGUIProperty Object)	Returns or sets the value of the property.

Delete Method (VMGUIProperty Object)

Object: [VMGUIProperty Object](#)

Prerequisites: None

Deletes the property.

Usage

```
VMGUIProperty.Delete() As Boolean
```

Arguments

None

Return Values

As Boolean. "True" if the property is deleted; "False" if the property is not deleted.

Name Property (VMGUIProperty Object)

Object: [VMGUIProperty Object](#)

Access: Read-Only

Prerequisites: None

Returns the name of the property.

Usage

VMGUIProperty.Name

Arguments

None

Return Values

String. The name of the property.

value Property (VMGUIProperty Object)

Object: [VMGUIProperty Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the value of the property.

Usage

```
VMGUIProperty.value = String
```

Arguments

None

Return Values

String. The value of the property.

VMSettingsCAE Object

The VMSettingsCAE object represents the CAE Interaction tab of the Settings dialog. Use this object to customize parameters when Variant Manager is open in the schematic design application.

The following section defines the methods and properties of the VMSettingsCAE object.

Table 14. VMSettingsCAE Object Methods and Properties

Method or Property	Description
GetMarkupColor Method (VMSettingsCAE Object)	Returns the Markup unplaced packages color values.
GetUnplaceColor Method (VMSettingsCAE Object)	Returns the Color unplaced packages color values.
LoadSchematicOnlyComponents Property (VMSettingsCAE Object)	Returns or sets the List schematic-only components option.
MarkupColor Property (VMSettingsCAE Object)	Returns the Markup unplaced packages VMColor object.
PackageUnplaceOption Property (VMSettingsCAE Object)	Returns or sets the unplaced packages options.
RemoveDanglingWires Property (VMSettingsCAE Object)	Returns or sets the Remove dangling wires option.
SetMarkupColor Method (VMSettingsCAE Object)	Sets the Markup unplaced packages color values.
SetUnplaceColor Method (VMSettingsCAE Object)	Sets the Color unplaced packages color values.
ShowSymbolReplacements Property (VMSettingsCAE Object)	Returns or sets the Show symbol replacements option.
UnplaceColor Property (VMSettingsCAE Object)	Returns the Color unplaced packages VMColor object.

GetMarkupColor Method (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Prerequisites: None

Returns the Markup unplaced packages color values.

Usage

```
VMSettingsCAE.GetMarkupColor(red As Long,  
                              green As Long,  
                              blue As Long)
```

Arguments

- red
A long that contains the red value (0-255)
- green
A long that contains the green value (0-255)
- blue
A long that contains the blue value (0-255)

GetUnplaceColor Method (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Prerequisites: None

Returns the Color unplaced packages color values.

Usage

```
VMSettingsCAE.GetUnplaceColor(red As Long,  
                                green As Long,  
                                blue As Long)
```

Arguments

- red
A long that contains the red value (0-255)
- green
A long that contains the green value (0-255)
- blue
A long that contains the blue value (0-255)

LoadSchematicOnlyComponents Property (VMSettingsCAE Object)

Prerequisites: None

Object: [VMSettingsCAE Object](#)

Access: Read/Write

Returns or sets the List schematic-only components option.

Usage

VMSettingsCAE.LoadSchematicOnlyComponents = True | False

Arguments

None

Return Values

Boolean. If True, checks the “List schematic-only components in Variant assignment grid” option on **Settings > CAE Interaction**. If False, unchecks the option.

MarkupColor Property (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Access: Read-Only

Prerequisites: None

Returns the Markup unplaced packages VMColor object.

Usage

VMSettingsCAE.MarkupColor

Arguments

None

Return Values

IMGCVMColor. The [VMColor Object](#).

PackageUnplaceOption Property (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the unplaced packages options.

Usage

VMSettingsCAE.PackageUnplaceOption = EVMCAEPackUnplace

Arguments

None

Return Values

EVMCAEPackUnplace. The package unplace option ([EVMCAEPackUnplace Enum](#)), Delete, Markup and Color.

Description

Defines how to handle the unplaced components or symbols in the variant view.

RemoveDanglingWires Property (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Remove dangling wires option.

Usage

VMSettingsCAE.RemoveDanglingWires = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

SetMarkupColor Method (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Prerequisites: None

Sets the Markup unplaced packages color values.

Usage

```
VMSettingsCAE.SetMarkupColor(ByVal red As Long,  
                              ByVal green As Long,  
                              ByVal blue As Long)
```

Arguments

- red
A long that contains the red value (0-255)
- green
A long that contains the green value (0-255)
- blue
A long that contains the blue value (0-255)

SetUnplaceColor Method (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Prerequisites: None

Sets the Color unplaced packages color values.

Usage

```
VMSettingsCAE.SetUnplaceColor(ByVal red As Long,  
    ByVal green As Long,  
    ByVal blue As Long)
```

Arguments

- red
A long that contains the red value (0-255)
- green
A long that contains the green value (0-255)
- blue
A long that contains the blue value (0-255)

ShowSymbolReplacements Property (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Show symbol replacements option.

Usage

VMSettingsCAE.ShowSymbolReplacements = True | False

Arguments

None

Return Values

Boolean. If True, checks the “Show symbol replacements for replaced packages” option on **Settings > CAE Interaction**. If False, unchecks the option.

UnplaceColor Property (VMSettingsCAE Object)

Object: [VMSettingsCAE Object](#)

Access: Read-Only

Prerequisites: None

Returns the Color unplaced packages VMColor object.

Usage

VMSettingsCAE.UnplaceColor

Arguments

None

Return Values

IMGCVMColor. The [VMColor Object](#).

VMSettingsGeneral Object

The VMSettingsGeneral object represents the General tab of the Setting dialog. Use this object to define the common settings for all Variant Manager clients.

The following section defines the VMSettingsGeneral object properties.

Table 15. VMSettingsGeneral Object Properties

Property	Description
CentLibPropsInGrid Property (VMSettingsGeneral Object)	Returns or sets selected Central Library symbol properties in the Variant Manager grid.
DisplaySelectedPropsOnly Property (VMSettingsGeneral Object)	Returns or sets the “Display selected properties only” option in the Variant Manager Settings dialog box. .
ReportTxtDelimiter Property (VMSettingsGeneral Object)	Returns or sets the Delimited text report: Field Separator option.
ReportViewExcel Property (VMSettingsGeneral Object)	Returns or sets Microsoft Excel report: View results option.
ReportViewHTML Property (VMSettingsGeneral Object)	Returns or sets the HTML document: View results option.
UnplacedKeyword Property (VMSettingsGeneral Object)	Returns or sets the Unplaced keyword option.

CentLibPropsInGrid Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets selected Central Library symbol properties in the Variant Manager grid.

Usage

```
VMSettingsGeneral.CentLibPropsInGrid = String
```

Arguments

None

Return Values

String. A string that contains the Central Library symbol properties to display in the grid.

Description

The selected Central Library symbol properties display as columns in the Variant Manager grid.

DisplaySelectedPropsOnly Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the “Display selected properties only” option in the Variant Manager Settings dialog box. .

Usage

VMSettingsGeneral.DisplaySelectedPropsOnly = True | False

Arguments

None

Return Values

True | False. If True, only the selected Central Library symbol properties appear in the Grid Customization list. If False, all Central Library symbol properties appear in the Grid Customization list.

Description

This option controls which Central Library symbol properties appear in the Grid Customization list in the Variant Manager Settings dialog box. It does not control the information displayed in the Variant Manager grid.

ReportTxtDelimiter Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Delimited text report: Field Separator option.

Usage

```
VMSettingsGeneral.ReportTxtDelimiter = String
```

Arguments

None

Return Values

String. The field separator to use in the delimited text document.

Description

Defines the field separator for the delimited text document (semicolon, comma, hyphen, etc.)

ReportViewExcel Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets Microsoft Excel report: View results option.

Usage

VMSettingsGeneral.ReportViewExcel = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Indicates when to open the Microsoft Excel report. If the option is set to “on”, a report automatically opens in Microsoft Excel after the report is generated. If set to “off”, the report saves to a file to open at a later time.

ReportViewHTML Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the HTML document: View results option.

Usage

VMSettingsGeneral.ReportViewHTML = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Indicates when to open the HTML report. If the option is set to “on”, the report automatically opens in HTML after the report is generated. If set to “off”, the report saves to a file to open at a later time.

UnplacedKeyword Property (VMSettingsGeneral Object)

Object: [VMSettingsGeneral Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Unplaced keyword option.

Usage

```
VMSettingsGeneral.UnplacedKeyword = String
```

Arguments

None

Return Values

String. A string that contains the keyword for the unplaced component.

Description

The keyword to use when the component is not placed in the design.

VMSettingsLibQuery Object

The VMSettingsLibQuery object represents the Library Query Setup tab of the Settings dialog. Use this object to setup the connection parameters for the library data source.

The following section defines the VMSettingsLibQuery object properties.

Table 16. VMSettingsLibQuery Object Properties

Property	Description
AliasPartNumberAttribute Property (VMSettingsLibQuery Object)	Returns or sets the alias part number attribute name.
ConfigFile Property (VMSettingsLibQuery Object)	Returns or sets the data source configuration file.
MaxResultsPerLib Property (VMSettingsLibQuery Object)	Returns or sets the Results /Library option.
MaxResultsTotal Property (VMSettingsLibQuery Object)	Returns or sets the Results total option.
PartNumberAttribute Property (VMSettingsLibQuery Object)	Returns or sets the Part number attr option.
RowFormatter Property (VMSettingsLibQuery Object)	Returns or sets the Query results format option.
SummaryInfoFormatter Property (VMSettingsLibQuery Object)	Returns or sets the Part number feedback line option.

The following example sets the options for library queries. This is equivalent to the Library Query Setup tab on the Settings dialog.

```
Dim LibQueryObj
Set LibQueryObj = SettingObj.LibQuery
LibQueryObj.ConfigFile = "E:\design\VidarOPN_DMS07_HSVSVR.dbc"
LibQueryObj.MaxResultsPerLib = 10
LibQueryObj.MaxResultsTotal = 20
LibQueryObj.PartNumberAttribute = "DEVICE"
LibQueryObj.RowFormatter = _
    "$(DEVICE)$(PKG_TYPE)$(TOLERANCE)$(VALUE)$(SYM)$(Library)"
LibQueryObj.SummaryInfoFormatter = "Part Name: $(DEVICE) at Library:" + _
    "$(Library), Geometry: $(PKG_TYPE)"
```


AliasPartNumberAttribute Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the alias part number attribute name.

Usage

```
VMSettingsLibQuery.AliasPartNumberAttribute = String
```

Arguments

None

Return Values

String. A string that contains the alias part number attribute name.

Description

This is the attribute name that identifies the alias part number in the library.

ConfigFile Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the data source configuration file.

Usage

```
VMSettingsLibQuery.ConfigFile = String
```

Arguments

None

Return Values

String. A string that contains the configuration (.dbc) file path.

Description

The path to the configuration (.dbc) file that stores and maintains the library connection setting.

MaxResultsPerLib Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Results /Library option.

Usage

```
VMSettingsLibQuery.MaxResultsPerLib = Integer
```

Arguments

None

Return Values

Integer. An integer that contains the number of results to return for each library.

Description

This defines the number of results to return for each library.

MaxResultsTotal Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Results total option.

Usage

VMSettingsLibQuery.MaxResultsTotal = Integer

Arguments

None

Return Values

Integer. An integer that contains maximum number of results in the query.

Description

This defines the maximum number of results the query returns.

PartNumberAttribute Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Part number attr option.

Usage

```
VMSettingsLibQuery.PartNumberAttribute = String
```

Arguments

None

Return Values

String. A string that contains the attribute name.

Description

This is the attribute name that identifies the part number in the library.

RowFormatter Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Query results format option.

Usage

```
VMSettingsLibQuery.RowFormatter = String
```

Arguments

None

Return Values

String. A string that contains the format of the query results.

Description

This defines the format of the Library Query Results Table.

Examples

```
LibQueryObj.RowFormatter =  
    "$ (DEVICE) $ (PKG_TYPE) $ (TOLERANCE) $ (VALUE) $ (SYM) $ (Library) "
```

SummaryInfoFormatter Property (VMSettingsLibQuery Object)

Object: [VMSettingsLibQuery Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Part number feedback line option.

Usage

VMSettingsLibQuery.SummaryInfoFormatter = String

Arguments

None

Return Values

String. A string that contains the format of the summary information displayed on the status line.

Description

This defines the how the part number properties display on the status line below the data entry grid.

Examples

```
LibQueryObj.SummaryInfoFormatter = "Part Name: $(DEVICE) at Library:" + _  
    "$ (Library), Geometry: $(PKG_TYPE)"
```

VMSettingsPADS Object

The VMSettingsPADS object represents the PADS tab of the Settings dialog. Use this object to setup the PADS options.

The following section defines the VMSettingsPADS object properties.

Table 17. VMSettingsPADS Object Properties

Property	Description
AttributePartName Property (VMSettingsPADS Object)	Returns or sets the Part name attr option.

AttributePartName Property (VMSettingsPADS Object)

Object: [VMSettingsPADS Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Part name attr option.

Usage

```
VMSettingsPADS.AttributePartName = String
```

Arguments

None

Return Values

String. The name of the part name property in PADS.

Description

This is the name of the part name property in iCDB in the PADS flow. The default value is "DEVICE".

VMSettingsPCB Object

The VMSettingsPCB object represents the PCB Interaction tab of the Settings dialog. Use this object to define the settings specific to the PCB layout and manufacturing applications.

The following section defines the VMSettingsPCB object properties.

Table 18. VMSettingsPCB Object Properties

Property	Description
DisplayNestedMechanicalCells Property (VMSettingsPCB Object)	Returns or sets the flag to display nested mechanical cells.
FablinkActionPrompt Property (VMSettingsPCB Object)	Returns or sets the Xpedition FabLink: Prompt user for action option.
FablinkDataGenerationMode Property (VMSettingsPCB Object)	Returns or sets the Xpedition FabLink data generation mode.
RemoveNested Property (VMSettingsPCB Object)	Returns or sets the Remove nested parts option.
UnplaceAssKeepOutline Property (VMSettingsPCB Object)	Returns or sets the Assembly display outline option.
UnplaceAssKeepPartnum Property (VMSettingsPCB Object)	Returns or sets the Assembly display part numbers option.
UnplaceAssKeepRefdes Property (VMSettingsPCB Object)	Returns or sets the Assembly display reference designator option.
UnplaceSilkKeepOutline Property (VMSettingsPCB Object)	Returns or sets the Silkscreen display outlines option.
UnplaceSilkKeepPartnum Property (VMSettingsPCB Object)	Returns or sets the Silkscreen display part numbers option.
UnplaceSilkKeepRefdes Property (VMSettingsPCB Object)	Returns or sets the Silkscreen display Reference designators option.
UnplaceSolderMask Property (VMSettingsPCB Object)	Returns or sets the Remove soldermask option.
UnplaceSolderPaste Property (VMSettingsPCB Object)	Returns or sets the Remove solderpaste option.

DisplayNestedMechanicalCells Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the flag to display nested mechanical cells.

Usage

VMSettingsPCB.DisplayNestedMechanicalCells = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

FablinkActionPrompt Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Xpedition FabLink: Prompt user for action option.

Usage

VMSettingsPCB.FablinkActionPrompt = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

This option defines the prompt for generating Xpedition FabLink data. If the option is set to “on”, the application prompts the end-user to generate Xpedition FabLink data. If set to “off”, the application does not prompt the end-user to generate Xpedition FabLink data.

FablinkDataGenerationMode Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Xpedition FabLink data generation mode.

Usage

```
VMSettingsPCB.FablinkDataGenerationMode = EVMFablinkDataGenerationMode
```

Arguments

None

Return Values

EVMFabLinkDataGenerationMode. An enumerate ([EVMFabLinkDataGenerationMode Enum](#)).

RemoveNested Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Remove nested parts option.

Usage

VMSettingsPCB.RemoveNested = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display nested parts from unplaced part padstacks. If the option is set to “on”, the parts nested inside an unplaced part do not display. If set to “off”, the nested parts remain with the unplaced part.

UnplaceAssKeepOutline Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Assembly display outline option.

Usage

VMSettingsPCB.UnplaceAssKeepOutline = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the assembly outline for unplaced parts. If the option is set to “on”, the assembly outline for the unplaced parts displays. If set to “off”, the assembly outline does not display.

UnplaceAssKeepPartnum Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Assembly display part numbers option.

Usage

```
VMSettingsPCB.UnplaceAssKeepPartnum = EVMOptionOnOffToggle
```

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the part number for unplaced parts on the assembly layer. If the option is set to “on”, the part number of the unplaced part displays on the assembly layer. If set to “off”, the part number does not display.

UnplaceAssKeepRefdes Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Assembly display reference designator option.

Usage

VMSettingsPCB.UnplaceAssKeepRefdes = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the reference designator for unplaced parts on the assembly layer. If the option is set to “on”, the reference designator for the unplaced parts displays on the assembly layer. If set to “off”, the reference designator does not display.

UnplaceSilkKeepOutline Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Silkscreen display outlines option.

Usage

VMSettingsPCB.UnplaceSilkKeepOutline = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the silkscreen outline for unplaced parts. If the option is set to “on”, the silkscreen outline displays for the unplaced parts. If set to “off”, the silkscreen outline does not display.

UnplaceSilkKeepPartnum Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Silkscreen display part numbers option.

Usage

VMSettingsPCB.UnplaceSilkKeepPartnum = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display part numbers for unplaced parts on the silkscreen layer. If the option is set to “on”, the part number of the unplaced part displays on the silkscreen layer. If set to “off”, the part number does not display.

UnplaceSilkKeepRefdes Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Silkscreen display Reference designators option.

Usage

VMSettingsPCB.UnplaceSilkKeepRefdes = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display reference designators for unplaced parts on the silkscreen layer. If the option is set to “on”, the reference designator of the unplaced part displays on the silkscreen layer. If set to “off”, the reference designator text does not display.

UnplaceSolderMask Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Remove soldermask option.

Usage

VMSettingsPCB.UnplaceSolderMask = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the soldermask for unplaced part padstacks. If the option is set to “on”, the soldermask of the unplaced part padstacks does not display. If set to “off”, the soldermask displays.

UnplaceSolderPaste Property (VMSettingsPCB Object)

Object: [VMSettingsPCB Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the Remove solderpaste option.

Usage

VMSettingsPCB.UnplaceSolderPaste = EVMOptionOnOffToggle

Arguments

None

Return Values

EVMOptionOnOffToggle. The [EVMOptionOnOffToggle Enum](#).

Description

Defines the option to display the solderpaste for unplaced part padstacks. If the option is set to “on”, the solderpaste of the unplaced part padstacks does not display. If set to “off”, the solderpaste displays.

VMSettings Object

The VMSettings object represents the Settings dialog. Use this object to return the settings objects for General, PCB Interaction, CAE Interaction, PADS Interaction and Library Query Setup.

The following section defines the VMSettings object properties.

Table 19. VMSettings Object Properties

Property	Description
CAE Property (VMSettings Object)	Returns the CAE settings object.
General Property (VMSettings Object)	Returns the General settings object.
LibQuery Property (VMSettings Object)	Returns the LibQuery settings object.
PADS Property (VMSettings Object)	Returns the PADS settings object.
PCB Property (VMSettings Object)	Returns the PCB settings object.

Example 3. Settings Example

The following code snippet, initializes the settings for the General, PCB, CAE and Library Query tabs.

```
Function setupSettings
    Dim settingsObj, toolSettingsObj
    Set settingsObj = appObj.Settings
    ' General Setting
    Set toolSettingsObj = settingsObj.General
    toolSettingsObj.ReportTxtDelimiter = ";"
    toolSettingsObj.ReportViewExcel = eVMOptionOn
    toolSettingsObj.ReportViewHTML = eVMOptionOn
    toolSettingsObj.UnplacedKeyword = "Unplaced"
    ' PCB Setting
    Set toolSettingsObj = settingsObj.PCB
    toolSettingsObj.FablinkActionPrompt = eVMOptionOn
    toolSettingsObj.FablinkDataGenerationMode = "On Save"
    toolSettingsObj.UnplaceAssKeepOutline = eVMOptionOn
    toolSettingsObj.UnplaceAssKeepPartnum = eVMOptionOn
    toolSettingsObj.UnplaceAssKeepRefdes = eVMOptionOn
    toolSettingsObj.UnplaceSilkKeepOutline = eVMOptionOn
    toolSettingsObj.UnplaceSilkKeepPartnum = eVMOptionOn
    toolSettingsObj.UnplaceSilkKeepRefdes = eVMOptionOn
    toolSettingsObj.UnplaceSolderMask = eVMOptionOn
    toolSettingsObj.UnplaceSolderPaste = eVMOptionOn
    ' CAE Setting
    Set toolSettingsObj = settingsObj.CAE
    Call toolSettingsObj.SetMarkupColor (255,255,255)'white
    Call toolSettingsObj.SetUnplaceColor (0, 200, 0) 'green
```

```
toolSettingsObj.PackageUnplaceOption = eVMCAEPackUnplaceMarkup
' Library Setting
Set toolSettingsObj = settingsObj.LibQuery
toolSettingsObj.ConfigFile = "C:\design\VidarOPN_DMS07_HSVSVR.dbc"
toolSettingsObj.MaxResultsPerLib = 10
toolSettingsObj.MaxResultsTotal = 20
toolSettingsObj.PartNumberAttribute = "DEVICE"
toolSettingsObj.RowFormatter = _
    "$(DEVICE)$$(PKG_TYPE)$$(TOLERANCE)$$(SYM)$$(Library)"
toolSettingsObj.SummaryInfoFormatter = "PartName: $(DEVICE) at Library
                                         $(Library), Geometry:
$(PKG_TYPE)"
End Function
```


CAE Property (VMSettings Object)

Object: [VMSettings Object](#)

Access: Read-Only

Prerequisites: None

Returns the CAE settings object.

Usage

VMSettings.CAE

Arguments

None

Return Values

IMGCVMSettingsCAE. The [VMSettingsCAE Object](#).

General Property (VMSettings Object)

Object: [VMSettings Object](#)

Access: Read-Only

Prerequisites: None

Returns the General settings object.

Usage

VMSettings.General

Arguments

None

Return Values

IMGCVMSettingsGeneral. The [VMSettingsGeneral Object](#).

LibQuery Property (VMSettings Object)

Object: [VMSettings Object](#)

Access: Read-Only

Prerequisites: None

Returns the LibQuery settings object.

Usage

```
VMSettings.LibQuery
```

Arguments

None

Return Values

IMGCVMSettingsLibQuery. The [VMSettingsLibQuery Object](#).

PADS Property (VMSettings Object)

Object: [VMSettings Object](#)

Access: Read-Only

Prerequisites: None

Returns the PADS settings object.

Usage

`VMSettings.PADS`

Arguments

None

Return Values

IMGCVMSettingsPADS. The [VMSettingsPADS Object](#).

PCB Property (VMSettings Object)

Object: [VMSettings Object](#)

Access: Read-Only

Prerequisites: None

Returns the PCB settings object.

Usage

VMSettings.PCB

Arguments

None

Return Values

IMGCVMSettingsPCB. The [VMSettingsPCB Object](#).

VMSymbolModification Object

The VMSymbolModification object represents a modified symbol. Use this object to access the modified symbol, its function and the modification operation. You can also set the operation type for modifying symbols.

The following section defines the methods and properties of the VMSymbolModification object.

Table 20. VMSymbolModification Object Methods and Properties

Method or Property	Description
ChangeOperation Method (VMSymbolModification Object)	Changes the operation for modifying symbols.
Delete Method (VMSymbolModification Object)	Deletes the symbol modification.
function Property (VMSymbolModification Object)	Returns the function for the modified symbol.
NewPartNumber Property (VMSymbolModification Object)	Returns the new part number for the symbol (Replace operations).
Operation Property (VMSymbolModification Object)	Returns the type of symbol modification.
symbol Property (VMSymbolModification Object)	Returns the modified symbol.

ChangeOperation Method (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Prerequisites: None

Changes the operation for modifying symbols.

Usage

```
VMSymbolModification.ChangeOperation(ByVal oper As EVMFMVOperation,  
    ByVal NewPartNumber As String,  
    [ByVal cellName As String]) As Integer
```

Arguments

- oper
The type of the operation ([EVMFMVOperation Enum](#)).
- NewPartNumber
A string that contains the new part number (Replace operation).
- cellName
(Optional) Reserved for future use.

Return Values

As Integer. "0" if the symbol is modified; "False" if the symbol is not modified.

Delete Method (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Prerequisites: None

Deletes the symbol modification.

Usage

```
VMSymbolModification.Delete() As Boolean
```

Arguments

None

Return Values

As Boolean. "True" if the modified symbol is removed; "False" if the modified symbol is not removed.

function Property (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Access: Read-Only

Prerequisites: None

Returns the function for the modified symbol.

Usage

`VMSymbolModification.function`

Arguments

None

Return Values

IMGCVMFunction. The [VMFunction Object](#).

NewPartNumber Property (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Access: Read-Only

Prerequisites: None

Returns the new part number for the symbol (Replace operations).

Usage

`VMSymbolModification.NewPartNumber`

Arguments

None

Return Values

String. The name of the new part number.

Description

Use [ChangeOperation](#) on page 211 method to change the part number.

Operation Property (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Access: Read-Only

Prerequisites: None

Returns the type of symbol modification.

Usage

VMSymbolModification.Operation

Arguments

None

Return Values

EVMFMVOperation. The [EVMFMVOperation Enum](#).

Description

Use [ChangeOperation](#) on page 211 method to change the symbol modification type.

symbol Property (VMSymbolModification Object)

Object: [VMSymbolModification Object](#)

Access: Read-Only

Prerequisites: None

Returns the modified symbol.

Usage

```
VMSymbolModification.symbol
```

Arguments

None

Return Values

IMGCVMSymbol. The [VMSymbol Object](#).

VMSymbol Object

The VMSymbol object represents a logic symbol. Use this object to access symbol attributes including, name, unique name (identifier), physical component, and symbol type. For hierarchical symbols, the object also accesses the parent symbol and the sub-symbols.

The following section defines the methods and properties of the VMSymbol object.

Table 21. VMSymbol Object Properties

Property	Description
Component Property (VMSymbol Object)	Returns the component for the symbol.
Name Property (VMSymbol Object)	Returns the name of the symbol.
NameAndPath Property (VMSymbol Object)	Returns the unique name of the symbol.
ParentSymbol Property (VMSymbol Object)	Returns the parent symbol.
Symbols Property (VMSymbol Object)	Returns the collection of sub-symbols for the symbol.
Type Property (VMSymbol Object)	Returns the symbol type (symbol, symbol block, reused block etc.).

Component Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the component for the symbol.

Usage

VMSymbol.Component

Arguments

None

Return Values

IMGCVMComponent. The [VMComponent Object](#) to which the symbol belongs. Returns Nothing if the symbol is not assigned to the component.

Name Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the name of the symbol.

Usage

VMSymbol.Name

Arguments

None

Return Values

String. The name of the symbol.

NameAndPath Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the unique name of the symbol.

Usage

```
VMSymbol.NameAndPath
```

Arguments

None

Return Values

String. The name and path of the symbol (symbol identification).

ParentSymbol Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the parent symbol.

Usage

`VMSymbol.ParentSymbol`

Arguments

None

Return Values

IMGCVMSymbol. The parent [VMSymbol Object](#) in the hierarchy.

Symbols Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of sub-symbols for the symbol.

Usage

```
VMSymbol.Symbols([ByVal filter As Variant])
```

Arguments

- filter

(Optional) The function name, [VMFunction Object](#), component name, or [VMComponent Object](#). If not specified, the property returns the symbols for the currently active function or component.

Return Values

IMGCVMSymbols. The sub-symbols as [VMSymbols Collection](#):

- With function filter, the property returns the collection of symbols in the function.
- With component filter, the property returns the collection of symbols that belong to the component.

Type Property (VMSymbol Object)

Object: [VMSymbol Object](#)

Access: Read-Only

Prerequisites: None

Returns the symbol type (symbol, symbol block, reused block etc.).

Usage

VMSymbol.Type

Arguments

None

Return Values

EVMSymbolType. The [EVMSymbolType Enum](#).

VMVariantGroup Object

The VMVariantGroup object represents a variant group. Use this object to assign and unassign variants to a variant group and to access the name, description and variants of the variant group.

The following section defines the methods and properties of the VMVariantGroup object.

Table 22. VMVariantGroup Object Methods and Properties

Method or Property	Description
Assign Method (VMVariantGroup Object)	Assigns a variant to the list of assigned variants.
AssignAll Method (VMVariantGroup Object)	Assigns all variants to the list of assigned variants.
Description Property (VMVariantGroup Object)	Returns or sets the description of the variant group.
Name Property (VMVariantGroup Object)	Returns or sets the name of the variant group.
UnAssign Method (VMVariantGroup Object)	Unassigns a variant in the list of assigned variants.
UnAssignAll Method (VMVariantGroup Object)	Unassigns all variants from the list of assigned variants.
Variants Property (VMVariantGroup Object)	Returns the collection of variants assigned to variant group.

Assign Method (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Prerequisites: None

Assigns a variant to the list of assigned variants.

Usage

```
VMVariantGroup.Assign(ByVal nameToAssign As Variant,  
    [ByVal position As Integer = -1]) As Integer
```

Arguments

- nameToAssign

A variant name or [VMVariant Object](#).

- position

(Optional) The position of the added variant. Specify -1 to add to the end of the list.

Return Values

As Integer. Value "1" if successful; value "0" if not executed.

AssignAll Method (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Prerequisites: None

Assigns all variants to the list of assigned variants.

Usage

```
VMVariantGroup.AssignAll() As Integer
```

Arguments

None

Return Values

As Integer. Reserved for future use.

Description Property (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the description of the variant group.

Usage

```
VMVariantGroup.Description = String
```

Arguments

None

Return Values

String. The description of the group.

Name Property (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the name of the variant group.

Usage

```
VMVariantGroup.Name = String
```

Arguments

None

Return Values

String. The name of the group.

UnAssign Method (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Prerequisites: None

Unassigns a variant in the list of assigned variants.

Usage

```
VMVariantGroup.UnAssign(ByVal nameToAssign As Variant) As Integer
```

Arguments

- nameToAssign

The variant name or [VMVariant Object](#).

Return Values

As Integer. Value “1” if successful; value “0” if not successful.

UnAssignAll Method (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Prerequisites: None

Unassigns all variants from the list of assigned variants.

Usage

```
VMVariantGroup.UnAssignAll() As Integer
```

Arguments

None

Return Values

As Integer. Reserved for future use.

Variants Property (VMVariantGroup Object)

Object: [VMVariantGroup Object](#)

Access: Read-Only

Prerequisites: None

Returns the collection of variants assigned to variant group.

Usage

`VMVariantGroup.Variants`

Arguments

None

Return Values

IMGCVMVariants. The [VMVariants Collection](#). Returns an empty collection if there are no assigned variants.

VMVariant Object

The VMVariant object represent a physical variant or a function managed variant (FMV). Use this object to access the attributes of a variant including the name, field number, description and variant type (physical or FMV). You can also work with the border properties, and set the options to include the variant in BOM reports and Xpedition FabLink data.

The following section defines the methods and properties of the VMVariant object.

Table 23. VMVariant Object Methods and Properties

Method or Property	Description
BOMInclude Property (VMVariant Object)	Returns or sets the option to include this variant in BOM reports.
BorderProperties Property (VMVariant Object)	Returns the property collection that defines the border.
DeleteProperty Method (VMVariant Object)	Deletes a border property.
Description Property (VMVariant Object)	Returns or sets the description of the variant.
FindProperty Method (VMVariant Object)	Finds and returns a border property.
FXEInclude Property (VMVariant Object)	Returns or sets the option to include the variant when generating Xpedition FabLink data.
Hyperlink Property (VMVariant Object)	Returns or sets the hyperlink description of the variant.
Name Property (VMVariant Object)	Returns or sets the name of the variant.
Number Property (VMVariant Object)	Returns or sets the field number of the variant.
PutProperty Method (VMVariant Object)	Adds a new border property.
Type Property (VMVariant Object)	Returns or sets the type of the variant (Physical or Function Managed).

BOMInclude Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the option to include this variant in BOM reports.

Usage

VMVariant.BOMInclude = True | False

Arguments

None

Return Values

True | False. “True” if the variant is included in the BOM reports; “False” if the variant is not included in the BOM reports.

BorderProperties Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read-Only

Prerequisites: None

Returns the property collection that defines the border.

Usage

```
VMVariant.BorderProperties
```

Arguments

None

Return Values

IMGCVMGUIProperties. The [VMGUIProperties Collection](#). The collection is empty if there is no border.

DeleteProperty Method (VMVariant Object)

Object: [VMVariant Object](#)

Prerequisites: None

Deletes a border property.

Usage

```
VMVariant.DeleteProperty(ByVal Name As String) As Boolean
```

Arguments

- Name

A string that contains the name of the property.

Return Values

As Boolean. "True" if the property deleted successfully; "False" if the property does not exist or there is an error.

Description Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the description of the variant.

Usage

```
VMVariant.Description = String
```

Arguments

None

Return Values

String. The description of the variant.

FindProperty Method (VMVariant Object)

Object: [VMVariant Object](#)

Prerequisites: None

Finds and returns a border property.

Usage

```
VMVariant.FindProperty(ByVal Name As String) As IMGCVMGUIProperty
```

Arguments

- Name

A string that contains the property name.

Return Values

As IMGCVMGUIProperty. The [VMGUIProperty Object](#). Returns Nothing if the property is not found.

FXEInclude Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the option to include the variant when generating Xpedition FabLink data.

Usage

VMVariant.FXEInclude = True | False

Arguments

None

Return Values

True | False. “True” if the variant is included when generating Xpedition FabLink data; “False” if the variant is not included when generating Xpedition FabLink data.

Hyperlink Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the hyperlink description of the variant.

Usage

```
VMVariant.Hyperlink = String
```

Arguments

None

Return Values

String. The hyperlink description of the variant.

Name Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the name of the variant.

Usage

```
VMVariant.Name = String
```

Arguments

None

Return Values

String. The name of the variant.

Number Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the field number of the variant.

Usage

```
VMVariant.Number = String
```

Arguments

None

Return Values

String. The number field of the variant.

PutProperty Method (VMVariant Object)

Object: [VMVariant Object](#)

Prerequisites: None

Adds a new border property.

Usage

```
VMVariant.PutProperty(ByVal Name As String,  
    ByVal value As String) As IMGCVMGUIProperty
```

Arguments

- Name
A string that contains the name of the property.
- value
A string that contains the value of the property.

Return Values

As IMGCVMGUIProperty. The [VMGUIProperty Object](#).

Type Property (VMVariant Object)

Object: [VMVariant Object](#)

Access: Read/Write

Prerequisites: None

Returns or sets the type of the variant (Physical or Function Managed).

Usage

```
VMVariant.Type = EVMVariantType
```

Arguments

None

Return Values

EVMVariantType. The type of the variant ([EVMVariantType Enum](#)).

Chapter 3

Variant Manager Collections

This section describes, in alphabetical order, the Variant Manager collection objects.

The following table includes summary information for each collection you can access in Variant Manager automation. To view the full description for a specific collection, click within the Collection column.

Table 24. Variant Manager Collections

Collection	Description
VMBlockModifications Collection	This section provides reference information for the supported properties of the VMBlockModifications Collection.
VMBorderPropModifications Collection	This section provides reference information for the supported properties of the VMBorderPropModifications Collection.
VMComponentModifications Collection	This section provides reference information for the supported properties of the VMComponentModifications Collection.
VMComponents Collection	This section provides reference information for the supported properties of the VMComponents Collection.
VMFunctionGroups Collection	This section provides reference information for the supported properties of the VMFunctionGroups Collection.
VMFunctions Collection	This section provides reference information for the supported properties of the VMFunctions Collection.
VMGUIProperties Collection	This section provides reference information for the supported properties of the VMGUIProperties Collection.
VMSymbolModifications Collection	This section provides reference information for the supported properties of the VMSymbolModifications Collection.
VMSymbols Collection	This section provides reference information for the supported properties of the VMSymbols Collection.
VMVariantGroups Collection	This section provides reference information for the supported properties of the VMVariantGroups Collection.
VMVariants Collection	This section provides reference information for the supported properties of the VMVariants Collection.

VMBlockModifications Collection

This section provides reference information for the supported properties of the VMBlockModifications Collection.

This is a collection of [VMBlockModification Objects](#).

Table 25. VMBlockModifications Collection Properties

Property	Description
Count Property (VMBlockModifications Collection)	Returns the number of VMBlockModification objects contained in the VMBlockModifications collection.
Item Property (VMBlockModifications Collection)	Returns a VMBlockModification object contained in the VMBlockModifications collection.

Count Property (VMBlockModifications Collection)

Collection: [VMBlockModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMBlockModification objects contained in the VMBlockModifications collection.

Usage

VMBlockModifications.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMBlockModifications Collection)

Collection: [VMBlockModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMBlockModification object contained in the VMBlockModifications collection.

Usage

VMBlockModifications.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMBlockModification. The [VMBlockModification Object](#). Fails when the index is not correct.

VMBorderPropModifications Collection

This section provides reference information for the supported properties of the VMBorderPropModifications Collection.

This is a collection of [VMBorderPropModification Objects](#).

Table 26. VMBorderPropModifications Collection Properties

Property	Description
Count Property (VMBorderPropModifications Collection)	Returns the number of VMBorderPropModification objects contained in the VMBorderPropModifications collection.
Item Property (VMBorderPropModifications Collection)	Returns a VMBorderPropModification object contained in the VMBorderPropModifications collection.

Count Property (VMBorderPropModifications Collection)

Collection: [VMBorderPropModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMBorderPropModification objects contained in the VMBorderPropModifications collection.

Usage

```
VMBorderPropModifications.Count
```

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMBorderPropModifications Collection)

Collection: [VMBorderPropModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMBorderPropModification object contained in the VMBorderPropModifications collection.

Usage

VMBorderPropModifications.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMBorderPropModification. The [VMBorderPropModification Object](#). Fails when the index is not correct.

VMComponentModifications Collection

This section provides reference information for the supported properties of the VMComponentModifications Collection.

This is a collection of [VMComponentModification Objects](#).

Table 27. VMComponentModifications Collection Properties

Property	Description
Count Property (VMComponentModifications Collection)	Returns the number of VMComponentModification objects contained in the VMComponentModifications collection.
Item Property (VMComponentModifications Collection)	Returns a VMComponentModification object contained in the VMComponentModifications collection.

Count Property (VMComponentModifications Collection)

Collection: [VMComponentModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMComponentModification objects contained in the VMComponentModifications collection.

Usage

VMComponentModifications.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMComponentModifications Collection)

Collection: [VMComponentModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMComponentModification object contained in the VMComponentModifications collection.

Usage

VMComponentModifications.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMComponentModification. The [VMComponentModification Object](#). Fails when the index is not correct.

VMComponents Collection

This section provides reference information for the supported properties of the VMComponents Collection.

This is a collection of [VMComponent Objects](#).

Table 28. VMComponents Collection Properties

Property	Description
Count Property (VMComponents Collection)	Returns the number of VMComponent objects contained in the VMComponents collection.
Item Property (VMComponents Collection)	Returns a VMComponent object contained in the VMComponents collection.

Count Property (VMComponents Collection)

Collection: [VMComponents Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMComponent objects contained in the VMComponents collection.

Usage

VMComponents.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMComponents Collection)

Collection: [VMComponents Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMComponent object contained in the VMComponents collection.

Usage

VMComponents.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMComponent. The [VMComponent Object](#). Fails when the index is not correct.

VMFunctionGroups Collection

This section provides reference information for the supported properties of the VMFunctionGroups Collection.

This is a collection of [VMFunctionGroup Objects](#).

Table 29. VMFunctionGroups Collection Properties

Property	Description
Count Property (VMFunctionGroups Collection)	Returns the number of VMFunctionGroup objects contained in the VMFunctionGroups collection.
Item Property (VMFunctionGroups Collection)	Returns a VMFunctionGroup object contained in the VMFunctionGroups collection.

Count Property (VMFunctionGroups Collection)

Collection: [VMFunctionGroups Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMFunctionGroup objects contained in the VMFunctionGroups collection.

Usage

```
VMFunctionGroups.Count
```

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMFunctionGroups Collection)

Collection: [VMFunctionGroups Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMFunctionGroup object contained in the VMFunctionGroups collection.

Usage

VMFunctionGroups.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMFunctionGroup. The [VMFunctionGroup Object](#). Fails when the index is not correct.

VMFunctions Collection

This section provides reference information for the supported properties of the VMFunctions Collection.

This is a collection of [VMFunction Objects](#).

Table 30. VMFunctions Collection Properties

Property	Description
Count Property (VMFunctions Collection)	Returns the number of VMFunction objects contained in the VMFunctions collection.
Item Property (VMFunctions Collection)	Returns a VMFunction object contained in the VMFunctions collection.

Count Property (VMFunctions Collection)

Collection: [VMFunctions Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMFunction objects contained in the VMFunctions collection.

Usage

VMFunctions.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMFunctions Collection)

Collection: [VMFunctions Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMFunction object contained in the VMFunctions collection.

Usage

`VMFunctions.Item (ByVal index As Variant)`

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMFunction. The [VMFunction Object](#). Fails when the index is not correct.

VMGUIProperties Collection

This section provides reference information for the supported properties of the VMGUIProperties Collection.

This is a collection of [VMGUIProperty Objects](#).

Table 31. VMGUIProperties Collection Properties

Property	Description
Count Property (VMGUIProperties Collection)	Returns the number of VMGUIProperty objects contained in the VMGUIProperties collection.
Item Property (VMGUIProperties Collection)	Returns a VMGUIProperty object contained in the VMGUIProperties collection.

Count Property (VMGUIProperties Collection)

Collection: [VMGUIProperties Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMGUIProperty objects contained in the VMGUIProperties collection.

Usage

VMGUIProperties.Count

Arguments

None

Return Values

Long. The number objects in the collection.

Item Property (VMGUIProperties Collection)

Collection: [VMGUIProperties Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMGUIProperty object contained in the VMGUIProperties collection.

Usage

```
VMGUIProperties.Item (ByVal index As Variant)
```

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMGUIProperty. The [VMGUIProperty Object](#). Fails when the index is not correct.

VMSymbolModifications Collection

This section provides reference information for the supported properties of the VMSymbolModifications Collection.

This is a collection of [VMSymbolModification Objects](#).

Table 32. VMSymbolModifications Collection Properties

Property	Description
Count Property (VMSymbolModifications Collection)	Returns the number of VMSymbolModification objects contained in the VMSymbolModifications collection.
Item Property (VMSymbolModifications Collection)	Returns a VMSymbolModification object contained in the VMSymbolModifications collection.

Count Property (VMSymbolModifications Collection)

Collection: [VMSymbolModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMSymbolModification objects contained in the VMSymbolModifications collection.

Usage

```
VMSymbolModifications.Count
```

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMSymbolModifications Collection)

Collection: [VMSymbolModifications Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMSymbolModification object contained in the VMSymbolModifications collection.

Usage

VMSymbolModifications.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMSymbolModification. The [VMSymbolModification Object](#). Fails when the index is not correct.

VMSymbols Collection

This section provides reference information for the supported properties of the VMSymbols Collection.

This is a collection of [VMSymbol Objects](#).

Table 33. VMSymbols Collection Properties

Property	Description
Count Property (VMSymbols Collection)	Returns the number of VMSymbol objects contained in the VMSymbols collection.
Item Property (VMSymbols Collection)	Returns a VMSymbol object contained in the VMSymbols collection.

Count Property (VMSymbols Collection)

Collection: [VMSymbols Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMSymbol objects contained in the VMSymbols collection.

Usage

VMSymbols.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMSymbols Collection)

Collection: [VMSymbols Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMSymbol object contained in the VMSymbols collection.

Usage

VMSymbols.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMSymbol. The [VMSymbol Object](#). Fails when the index is not correct.

VMVariantGroups Collection

This section provides reference information for the supported properties of the VMVariantGroups Collection.

This is a collection of [VMVariantGroup Objects](#).

Table 34. VMVariantGroups Collection Properties

Property	Description
Count Property (VMVariantGroups Collection)	Returns the number of VMVariantGroup objects contained in the VMVariantGroups collection.
Item Property (VMVariantGroups Collection)	Returns a VMVariantGroup object contained in the VMVariantGroups collection.

Count Property (VMVariantGroups Collection)

Collection: [VMVariantGroups Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMVariantGroup objects contained in the VMVariantGroups collection.

Usage

```
VMVariantGroups.Count
```

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMVariantGroups Collection)

Collection: [VMVariantGroups Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMVariantGroup object contained in the VMVariantGroups collection.

Usage

VMVariantGroups.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMVariantGroup. The [VMVariantGroup Object](#). Fails when the index is not correct.

VMVariants Collection

This section provides reference information for the supported properties of the VMVariants Collection.

This is a collection of [VMVariant Objects](#).

Table 35. VMVariants Collection Properties

Property	Description
Count Property (VMVariants Collection)	Returns the number of VMVariant objects contained in the VMVariants collection.
Item Property (VMVariants Collection)	Returns a VMVariant object contained in the VMVariants collection.

Count Property (VMVariants Collection)

Collection: [VMVariants Collection](#)

Access: Read-Only

Prerequisites: None

Returns the number of VMVariant objects contained in the VMVariants collection.

Usage

VMVariants.Count

Arguments

None

Return Values

Long. The number of objects in the collection.

Item Property (VMVariants Collection)

Collection: [VMVariants Collection](#)

Access: Read-Only

Prerequisites: None

Returns a VMVariant object contained in the VMVariants collection.

Usage

VMVariants.Item (ByVal index As Variant)

Arguments

- index

An integer or long that indicates the item to return.

Return Values

VMVariant. The [VMVariant Object](#). Fails when the index is not correct.

Chapter 4

Variant Manager Enumerated Types

This section provides reference information for the Variant manager enumerated types.

[Variant Manager Enumerated Types Summary](#)

Variant Manager Enumerated Types Summary

The table below includes summary information for each enumerated type you can access with Variant Manager Automation.

To view the full description for a specific enumerate, click within the Enumerated Type column.

Table 36. Variant Manager Enumerated Types

Enumerated Type	Description
EVMBOMType Enum	BOM list type constants.
EVMCAEPackUnplace Enum	Constants for delete, markup, color, and dashed-linestyle unplaced packages in the schematic design.
EVMComponentType Enum	Constants for component types (component, jumper and mechanical).
EVMErrCode Enum	Error codes constants.
EVMEventOperation Enum	Variant event operation constants.
EVMFabLinkDataGenerationMode Enum	Xpedition FabLink data generation constants.
EVMFMVOperation Enum	FMV operation constants.
EVMGRGUICAEPackUnplace Enum	Constants for delete, markup, color, and dashed-linestyle unplaced packages in the schematic design.
EVMOperation Enum	Modification/operation constants.
EVMOptionOnOffToggle Enum	Toggle, On and Off constants.
EVMPADSOutputErrorCode Enum	PADS output error code constants.
EVMReportFormat Enum	BOM report format constants. (Delimited text, Excel, formatted text, HTML, or undefined.)
EVMSymbolType Enum	Symbol type constants.
EVMVariantType Enum	Variant types. (FMV or Physical).

EVMBOMType Enum

Prerequisites: None.

BOM list type constants.

Usage

EVMBOMType

Arguments

- eVMBOMPartList
The numerical value for this constant is 0.
- eVMBOMRefDesList
The numerical value for this constant is 1.

EVMCAEPackUnplace Enum

Prerequisites: None.

Constants for delete, markup, color, and dashed-linestyle unplaced packages in the schematic design.

Usage

EVMCAEPackUnplace

Arguments

- eVMCAEPackUnplaceColor
The numerical value for this constant is 3.
- eVMCAEPackUnplaceDashLS
The numerical value for this constant is 4
- eVMCAEPackUnplaceDelete
The numerical value for this constant is 1.
- eVMCAEPackUnplaceMarkup
The numerical value for this constant is 2.

EVMComponentType Enum

Prerequisites: None.

Constants for component types (component, jumper and mechanical).

Usage

EVMComponentType

Arguments

- eVMComponent
The numerical value for this constant is 0.
- eVMJumper
The numerical value for this constant is 4.
- eVMMechanicalCell
The numerical value for this constant is 1.

EVMErrCode Enum

Prerequisites: None.

Error codes constants.

Usage

EVMErrCode

Arguments

eVMErrCannotBeEmpty	2147220929 (&H8004023F).	Empty argument name not allowed.
eVMErrCannotChangeDefault Group	2147220741 (&H800402FB).	Cannot change property, default function or default variant group.
eVMErrCannotChangeVariant Mode	2147220922 (&H80040426)	Cannot change variant mode.
eVMErrCannotChangeVariant Type	2147220936 (&H80040238).	Cannot change physical variant type. This is allowed only in FMV mode.
eVMErrCannotCreate	2147220930 (&H8004023E).	Cannot create object.
eVMErrCannotReplaceOnBlock	2147220923 (&H80040245)	Cannot replace on block.
eVMErrCollIndexOutOfRange	2147220938 (&H80040236).	Out of range collection index.
eVMErrCollInvalidIndex	2147220937 (&H80040237).	Invalid index argument. Only long and integer allowed.
eVMErrDMSDesktopLibraryNot Found	2147220929 (&H8004023F).	
eVMErrDMSDesktopNotConn ected	2147220740 (&H800402FC)	
eVMDMSDesktopPartNotFo und	2147220739 (&H800402FD)	
eVMErrDoesNotExist	2147220931 (&H8004023D).	Object does not exist, or incorrect argument name.
eVMErrExists	2147220932 (&H8004023C).	Object with the same name already exists.
eVMErrFileMustBeChangedIn Designer	-2147220735 (&H80040301)	The DBC file and production library limitation must be set in Designer.

eVMErrFileNotSupported	-2147220736 (&H80040300)	The DBC file requires a production library limitation, which is not supported in designs of this flow type.
eVMErrInvalidObject	2147220941 (&H80040233).	Object is not valid.
eVMErrInvalidParamNameOr Object	2147220935 (&H80040239).	Argument is incorrect name or object.
eVMErrNoAcceptedName	2147220927 (&H80040241).	Incorrect name.
eVMErrNoModsInReadOnly	2147220939 (&H80040235).	Object is read-only, cannot modify.
eVMErrNotEquivalentParts	2147220926 (&H80040242).	Parts are not equivalent, cannot replace.
eVMErrObjectAlreadyAssignedInGroup	2147220933 (&H8004023B).	Variant or function is already assigned to the variant or function group.
eVMErrObjectNotAssignedInGroup	2147220934 (&H8004023A).	Variant or function is not assigned to the group, therefore cannot unassign.
eVMErrTooLong	2147220928 (&H80040240).	String argument is too long.
eVMErrUndefinedError	2147220940 (&H80040234).	Undefined error.

EVMEvOperation Enum

Prerequisites: None.

Variant event operation constants.

Usage

EVMEvOperation

Arguments

- eVMOperationAssignmentChange
Assignment changed (for example: function assigned to variant; function assigned to function group; variant assigned to variant group). The numerical value for this constant is 11 (&HB).
- eVMOperationAssignmentDelete
Assignment deleted. The numerical value for this constant is 14 (&HE).
- eVMOperationChange
Object changed. The numerical value for this constant is 1.
- eVMOperationDelete
Object deleted. The numerical value for this constant is 3.
- eVMOperationNew
New object added. The numerical value for this constant is 2.

EVMFabLinkDataGenerationMode Enum

Prerequisites: None.

Xpedition FabLink data generation constants.

Usage

EVMFabLinkDataGenerationMode

Arguments

- eVMFXENoGenerate

Assignment does not generate data. The numerical value for this constant is 0.

- eVMFXEOnExit

Assignment generates data upon exiting. The numerical value for this constant is 2.

- eVMFXEOnRequest

Assignment generates data when the Xpedition FabLink icon on the toolbar is selected. The numerical value for this constant is 3.

- eVMFXEOnSave

Assignment generates data during a Save. The numerical value for this constant is 1.

EVMFMVOperation Enum

Prerequisites: None.

FMV operation constants.

Usage

EVMFMVOperation

Arguments

- eVMFMVOperExclude
Modification is excluded. The numerical value for this constant is 2.
- eVMFMVOperInclude
Modification is included. The numerical value for this constant is 1.
- eVMFMVOperIncReplace
Modification is included and replaced. The numerical value for this constant is 5.
- eVMFMVOperNone
The numerical value for this constant is 0.
- eVMFMVOperReplace
Modification is replaced. The numerical value for this constant is 4.

EVMGRGUICAEPackUnplace Enum

Prerequisites: None.

Constants for delete, markup, color, and dashed-linestyle unplaced packages in the schematic design.

Usage

EVMGRGUICAEPackUnplace

Arguments

- evmgrGUICAEPackUnplaceColor
The numerical value for this constant is 3.
- evmgrGUICAEPackUnplaceDashLS
The numerical value for this constant is 4
- evmgrGUICAEPackUnplaceDelete
The numerical value for this constant is 1.
- evmgrGUICAEPackUnplaceMarkup
The numerical value for this constant is 2.

EVMOperation Enum

Prerequisites: None.

Modification/operation constants.

Usage

EVMOperation

Arguments

- eVMOperNone

The numerical value for this constant is 0.

- eVMOperReplace

Component is replaced. The numerical value for this constant is 2.

- eVMOperUnplace

Component is unplaced. The numerical value for this constant is 1.

EVMOptionOnOffToggle Enum

Prerequisites: None.

Toggle, On and Off constants.

Usage

EVMOptionOnOffToggle

Arguments

- eVMOptionOff
Off. The numerical value for this constant is 0.
- eVMOptionOn
On. The numerical value for this constant is 1.
- eVMOptionToggle
Toggle between on and off. The numerical value for this constant is 2.

EVMPADSOutputErrorCode Enum

Prerequisites: None

PADS output error code constants.

Usage

EVMPADSOutputErrorCode

Arguments

- eVMFileNotExist
The numerical value for this constant is 1.
- eVMMissingOrEmptyConfigFile
The numerical value for this constant is 2.
- eVMObjectsNotInitialized
The numerical value for this constant is 3.
- eVMSuccess
The numerical value for this constant is 0.

EVMReportFormat Enum

Prerequisites: None.

BOM report format constants. (Delimited text, Excel, formatted text, HTML, or undefined.)

Usage

EVMReportFormat

Arguments

- eVMDelimitedText
Text delimited. The numerical value for this constant is 3.
- eVMExcel
Excel. The numerical value for this constant is 4.
- eVMFormattedText
Formatted text. The numerical value for this constant is 1.
- eVMHTML
HTML. The numerical value for this constant is 2.
- eVMUndefined
Undefined. The numerical value for this constant is 0.

EVMSymbolType Enum

Prerequisites: None.

Symbol type constants.

Usage

EVMSymbolType

Arguments

- eVMSymbol
Simple symbol. The numerical value for this constant is 1.
- eVMSymbolBlock
Block symbol. The numerical value for this constant is 2.
- eVMSymbolReusedBlock
Reusable block symbol. The numerical value for this constant is 3.
- eVMSymbolReusedBlockFlattened
Flattened reusable block symbol. The numerical value for this constant is 4.
- eVMSymbolBorder
Symbol border. The numerical value for this constant is 5.
- eVMSymbolUndefined
Undefined symbol. The numerical value for this constant is 0.

EVMVariantType Enum

Prerequisites: None.

Variant types. (FMV or Physical).

Usage

EVMVariantType

Arguments

- eVMFMVVariant

Variant is function managed variant type. The numerical value for this constant is 1.

- eVMPhysicalVariant

Variant is physical variant type. The numerical value for this constant is 2.

Third-Party Information

Details on open source and third-party software that may be included with this product are available in the `<your_software_installation_location>/legal` directory.