

Overview of Adoption for Automation

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the **3DEXPERIENCE** platform.

This article describes or points to articles describing the main new concepts and technologies you need to understand before looking at your Automation macros. Then a change methodology is explained along with where to find the appropriate information.

What Is New with **3DEXPERIENCE** Platform Automation?

Automation reflects the changes brought to **3DEXPERIENCE** platform compared to V5.

The main change in the data model consists in replacing the document-centric model relying on files with a product/representation model relying on a database and a client/server architecture. Main characteristics are:

- There are no Part, Product, Drawing, Material, or Analysis documents and files any longer. They are replaced with product references and instances.
- All data models are specializations of the same PLM Core model that enables an non-semantic server-layer code to perform robust query and save operations on stateless servers, removing data mapping problems and data loss.
- The Product model, as well as the other data models, is a unique entity-relation based model designed to reduce the storage size in the database, to allow for large assemblies in sessions and for perfect symmetry without data loss when back and forth loading and storing data.
- Parts support the multi-representation thanks to isolation between the part reference and its shape.
- Each PLM entity is assigned a unique identifier whatever its type supporting any PLM operations, such as versioning, configuration, or replication.

Other changes are related to the user interface that offers the Bar and the Compass in addition to the 3D compass, for easy search, navigation, and real-time collaboration.

How to Adopt **3DEXPERIENCE** Platform Automation?

Before modifying macros to take advantage of **3DEXPERIENCE** platform architecture, a minimal understanding of PLM concepts, PLM objects customization and products is required and can be found in the product documentation. The following steps should then be considered:

1. Choose target languages for **3DEXPERIENCE** platform macros.
2. Understand main modifications that will have to be performed.
3. Assess **3DEXPERIENCE** platform functional level to understand which macros will run on the current level.
4. Define the structure of libraries that will allow **macro persistency**, create any required library and macro, and initialize macros with V5 content.
5. Review all macros. Modify what is required and test code using the *Details of modifications on V5 modelers* part of the *Adoption* section as a reference.

Those steps are described below in more details.

Choose Target Language for **3DEXPERIENCE** Platform Macros

Of course CATScript, VBScript and VBA are still supported.

Scripting languages supported in V5 are still supported: CATScript, Visual Basic for Application and Visual Basic Scripting Edition (VBScript) can be used, detailed levels are provided in the product Program Directory. **3DEXPERIENCE** platform now integrates VSTA, meaning that C# and VB.NET are additional scripting languages to choose from. So it is possible either to keep the choices that were made in V5 or to take advantage of **3DEXPERIENCE** platform adoption to review those choices.

For example CATScript files content is dynamically preprocessed into VBScript before being executed. This runtime cost could be avoided by removing any typing information from the V5 CATScript macros and storing the result in a catvbs macro. Migrating macros to C# or VB.NET could also be considered in the perspective of complying to a development language corporate strategy.

Understand Main Modifications

This approach is based on main V5 scenarios commonly used when creating macros and aims to understand which are the main modifications that will be required. The articles provided in *Main Applications Patterns* part of the *Adoption* section should be read first, they address:

- Automating Access to Settings.
- Working with Objects.
- Bringing Objects into Session and Saving objects.
- Accessing Objects in Session.
- Creating Objects.
- Accessing Objects Through Generic Methods.

It is also recommended to read the article [Product Changes](#) in the *Details of modifications on V5 modelers* part of the *Adoption* section.

Assess **3DEXPERIENCE** Platform Functional Level

All V5 object and methods may not be available in the current **3DEXPERIENCE** platform level. As there is no explicit compilation of macro code for languages available in V5, this cannot be fully performed before the code modification phase but high level hints can be found:

- In the *Object Reference* part of the *Reference* section that lists available **3DEXPERIENCE** platform objects
- In the *Details of modifications on V5 modelers* part of the *Adoption* section that described for V5 modelers available in the current **3DEXPERIENCE** platform level, which thinner level functionality may not be available.

Define Macros Persistency

Macros are saved and accessed in macro libraries stored in the PLM server database. This allows easy access from any client seat to corporate macros. Macro libraries must so be created first, typically their number and names will reflect language used by stored macros and any kind of logical gathering (processes where they should be used, functional domains, etc.).

For each V5 macro that will be processed, a **3DEXPERIENCE** platform macro should be created in one of the libraries, and the code should copy/pasted from the V5 document they were saved in.

- CATScript and catvbs files can be edited using any text editor.

- CAVBA files must be read first using a V5 session and should be exported as .bas files using the embedded VBA editor. This .bas file will then be imported using the VBA editor associated with a 3DEXPERIENCE platform session.
- Macros stored in non macro-specific V5 documents like CATPart or CATDrawing files, cannot be saved in their hosting representations in 3DEXPERIENCE platform. You can:
 - Store them in the database: import the V5 file. A dedicated representation is created to contain the macro. You can save this macro in a PLM macro library corresponding to the macro language (CATScript or VBScript). You can then recall the macro from its PLM macro library using the Macros dialog box of the Macro command in the Tools menu. Because the V5 automation model is not compliant with the 3DEXPERIENCE platform one, most of V5 macros cannot be used "As Is". To prevent problems, the FBDI process adds a string at the beginning of all imported macros to make them non-executable. Consult the article *Macros File-based Design Import* in the product documentation for more information.
 - Store them in files: extract them from the V5 documents using a V5 session first, and save them as text in text files. The content of this text file can then be pasted in a 3DEXPERIENCE platform macro edited in a 3DEXPERIENCE platform session. V5 Documents do not need to be modified, the File Based Design Import mechanism (FBDI), that allows you to import V5 documents into the PLM server, will remove the macros code when the V5 document will be imported in the PLM server database.

Notes:

1. Documents that were dedicated to store macros in V5 (files with the .CATBVA, .CATScript or .catvbs extension) are not handled by FBDI.
2. Macro code stored in Knowledgeware features is kept when importing the document containing those features. A warning pop-up is however added. Consult the article *Knowledgeware File-based Design Import* in the product documentation for more information.

Review, Modify, and Test Code

Once main modifications have been understood, persistency defined and 3DEXPERIENCE platform macro code initialized with V5 content, actual code modification and test can take place. Here again the *Details of modifications on V5 modelers* part of the *Adoption* section should be used. It describes, for each V5 modeler available, which objects and methods have been changed and how.

Each macro should at least be reviewed and any usage of Documents collection, Document object or any object whose types derives from Document should be replaced.

Note: A V5 macro should never be considered as acceptable for 3DEXPERIENCE platform if it has not at least been reviewed.

What Is Not Possible in Current Level?

Functionality involving links from representations to PLM object are not supported:

- Creation of contextual design.

Automating Settings

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open 3DEXPERIENCE platform PLM infrastructure. It explains how to transform an application that was automating access to settings to make it 3DEXPERIENCE platform compliant.

The way to access settings through Automation is simplified and can now be performed most of the time using the generic methods of the **SettingRepository** object relying on the XML-based description of the setting repositories and attributes.

V5 setting attributes are each managed using a dedicated read/write property and two dedicated methods: one for locking/unlocking the setting attribute, and one for retrieving information about it. For each setting attribute, in addition to its name and role, the tab page where it is accessed interactively, you then need to know or find its setting controller object and the way to retrieve it, and the property and method names to manage the setting attribute.

3DEXPERIENCE platform setting attributes are declared in XML files and managed using the set of six generic methods of the single **SettingRepository** object:

- GetAttr and PutAttr to retrieve and set the value of a simple type setting attribute.
- GetAttrArray and PutAttrArray to retrieve and set the value of an array type setting attribute.
- GetAttrInfo to retrieve information about a setting attribute: is it administrated, locked, modified at the current administration level?
- SetAttrLock to lock or unlock a setting attribute.

To access a 3DEXPERIENCE platform setting attribute, you just need to know its name and in which setting repository it is stored.

For example, the following pattern can be applied to retrieve a setting attribute value:

V5 Automation	3DEXPERIENCE platform Automation
<pre>' Accessing the setting controller dedicated to ' the global session settings tab page Dim cSettingControllers As SettingControllers Set cSettingControllers = CATIA.SettingControllers Dim oGeneralSessionSettingAtt As GeneralSessionSettingAtt Set oGeneralSessionSettingAtt = cSettingControllers.Item("CATCafGeneralSessionSettingCtrl")</pre>	<pre>' Accessing a generic setting controller for ' the global session settings tab page Dim cSettingControllers As SettingControllers Set cSettingControllers = CATIA.SettingControllers Dim oSettingRepository As SettingRepository Set oSettingRepository = cSettingControllers.Item("GeneralGeneral")</pre>
<pre>' Determining if Drag & Drop is enabled ' to Copy/Paste using a specific property Dim bDragDropForCPPEnabled bDragDropForCPPEnabled = oGeneralSessionSettingAtt.DragDrop</pre>	<pre>' Determining if Drag & Drop is enabled ' to Copy/Paste using the generic method Dim bDragDropForCPPEnabled bDragDropForCPPEnabled = oSettingRepository.GetAttr("DragAndDrop")</pre>

In this example, the V5 drag & drop ability setting attribute is managed by a specific property (DragDrop) of the specific setting controller object (GeneralSessionSettingAtt) managing the underlying setting repository. Another setting attribute is managed using a different property of possibly a different setting controller if the setting attribute belongs to a different setting repository. With 3DEXPERIENCE platform, the setting repositories are handled using a single object (SettingRepository), and the value of any setting attribute of a simple type, that is, that is not an array, is retrieved thanks to the GetAttr method.

For more information, consult the [SettingController Object](#), that describes how to retrieve a **SettingRepository** object and details how to use it, and the [Setting Repository Reference](#) that describes and lists the setting repositories along with their setting attributes.

Note that the following V5 setting controller objects still exist in 3DEXPERIENCE platform. The associated setting repositories cannot thus be managed using the generic setting controller:

- [DLNameSettingAtt](#)
- [DynLicenseSettingAtt](#)
- [LicenseSettingAtt](#)

- [MacrosSettingAtt](#)

In addition, note that the **FileAccessStatisticsSettingAtt** object is removed in **3DEXPERIENCE** platform, since no files are used any longer.

Working with Objects

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM infrastructure. It focuses on generic methods inherited by most objects from the **AnyObject** object or the **Collection** object:

The Parent Property

As in V5, **3DEXPERIENCE** platform objects are aggregated by a parent object, itself being aggregated by its parent, thus creating a tree whose root is the **Application** object. The parent of an object can be obtained using its **Parent** property. Depending on the object, the behavior of the **Parent** property may have changed in **3DEXPERIENCE** platform. As a general rule:

- The behavior of non persistent objects (windows, viewers, etc.) does not change.
- Objects that persist inside **3DEXPERIENCE** platform representation references like part or drawing data keep the same behavior, except when reaching the layer of PLM objects. For example:

V5 Automation	3DEXPERIENCE platform Automation
' Accessing the parent of a Part object	' Accessing the parent of a Part object
Set oObject = oPart.Parent	Set oObject = oPart.Parent
' This parent is a PartDocument object	' This parent is a VPMRepReference object
msgbox typeName (oObject)	msgbox typeName (oObject)
- PLM entities, meaning objects that can be queried through PLM queries in **3DEXPERIENCE** platform, like the V5 **Product** object, are replaced, even if the concept is kept. The way they are aggregated, and so the behavior of the **Parent** property, may have changed. Consult the *Details of modifications on V5 modelers* part of this *Adoption* section for more information.

The Name property

The **Name** property allows you to identify objects, for example inside collections. Here again only the behavior of V5 objects that become **3DEXPERIENCE** platform **PLMEntity** objects is changed, mainly because **PLMEntity** objects are identified by a customizable set of PLM attribute values, called the *PLM Identifier Set*, and not anymore by a simple string. The most common usages of the **Name** property are described below.

Adding Attributes to an Object

Very often the name of a V5 document, for example, is made up of a concatenation of attributes, allowing you to classify and retrieve objects:

```
<part number>_<responsible>.CATPart
```

In **3DEXPERIENCE** platform those attributes are typical PLM attributes, defined on the PLM representation reference object.

One specific type of attribute implemented this way is the extension of the document: retrieving all **Product** documents in a folder in V5 is performed through a query on product objects in **3DEXPERIENCE** platform.

Note: the **Path** property of the V5 **Document** object is often used the same way.

Displaying a User Understandable Name

For example, in a macro that performs checks on V5 documents, when a problem is discovered, the message must allow the end user to understand to which object it applies. In V5, the name of the document was the obvious information to provide. In **3DEXPERIENCE** platform this information could be part of, or the whole PLM Identifier Set and maybe specific attributes. In any case this heavily depends on the customization applied to PLM metadata, that itself aims to implement corporate processes.

The GetItem Function

The **GetItem** function allows you to retrieve an aggregated object from its name particularly when the aggregation is dynamic. For example, a **Part** object does not aggregate a **Layout2DRoot** object if the *2D Layout for 3D Design* application was not used. Otherwise, the **Layout2DRoot** object is retrieved this way:

V5 Automation

```
' Accessing the 2D Layout root
Dim o2DRoot As Layout2DRoot
Set o2DRoot = CATIA.ActiveDocument.Part.GetItem("CATLayout2DRoot")
```

As for the **Parent** property, this may have changed depending on the objects with which it was used when PLM objects are involved. Consult the *Details of modifications on V5 modelers* part of this *Adoption* section for more information.

Bringing Objects into Session And Saving Objects

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM infrastructure. It explains how the V5 document centric model and more precisely the **Documents** collection has been replaced to accommodate database persistency.

In V5 the **Document** object is the key object that allows persistency. Through the **Documents** collection objects, new **Document** objects can be created as a repository for persistent data, modifications performed in session can be saved on disk, and existing **Document** objects loaded data into session.

In **3DEXPERIENCE** platform persistent data is stored in the PLM server, either as a PLM object and its attributes and links or as the black box content of a specific PLM objects: the representation references.

This article explains how to bring objects into session, save objects, and related considerations. Creation of objects is described in the [Creating Objects](#) article.

Bringing Objects into Session

In V5, documents are stored on file systems and identified through a path. This path, including the document name, is often used to classify and add attributes on the document. The **Open** method of the **Documents** collection allows you to load a document in memory from its path. The content of the document can then be edited or browsed.

In **3DEXPERIENCE** platform, the PLM server contains PLM entities that can be queried against a set of PLM attributes, using the **SearchService** service object. The result of a query is a set of PLM entities that must be opened using the **PLMOpen** method of the **PLMOpenService** object before being browsed or edited through specific editors.

V5 Automation

```
' A Product document is known by its path
sIdentifier="E:/myDocument.CATProduct"
```

```
' Its content is loaded through the Documents collection
Set oProductDocument = CATIA.Documents.Open("sIdentifier")
```

3DEXPERIENCE platform Automation

```
' Get the SearchService object
Dim oSearchService As SearchService
Set oSearchService = CATIA.GetSessionService("Search")

' Create a query using the DatabaseSearch object
Dim oDBSearch As DatabaseSearch
Set oDBSearch = oSearchService.DatabaseSearch

' Set the type of object to query, a criterion, and trigger the search
oDBSearch.BaseType = "VPMReference"
oDBSearch.AddEasyCriteria "PLM_ExternalID", "Product"

' Launch the search
oSearchService.Search

' Retrieve the listed entities
Dim cPLMEntities As PLMEntities
Set cPLMEntities = oDBSearch.Results

' Get the PLMOpen service & Open the first retrieved entity
Set oPLMOpenService = CATIA.GetSessionService("PLMOpenService")
Set oEditor = oPLMOpenService.PLMOpen (cPLMEntities.Item(1))
```

Note: depending on the query, the list of retrieved PLM entities may contain more than one object.

Saving Objects

In **3DEXPERIENCE** platform, the save operation applies to all PLM objects opened in the active editor and is performed through the **PLMPropagate** method of the **PLMPropagateService** object.

V5 Automation

```
' Saving the active Document ' Saving All data in Session
CATIA.ActiveDocument.Save    CATIA.GetSessionService("PLMPropagateService").PLMPropagate
```

3DEXPERIENCE platform Automation

Using the File Selection Box

The **FileSelectionBox** property of the **Application** object allows to display a window where the path of a file can be selected. In **3DEXPERIENCE** platform, this can still be used to retrieve documents stored on a file system but PLM entities must be retrieved either through the **SearchService** service object, or using the interactive **Search** command.

Accessing and Editing Objects in Session

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM infrastructure. It explains how the V5 document centric model is modified to provide access to objects in session.

Accessing the Document and the Editor

In V5 the **Document** object gives access to persistent data loaded in session. This notion has been replaced in **3DEXPERIENCE** platform by an **Editor** object. In the Model View Controller paradigm, the editor plays the Controller role. The editor federates all the objects that can be interactively edited in the same window. As for the V5 **Document** object, the application object aggregates an active **Editor** object and a collection of **Editor** objects. The following illustrates the modifications required to access a **3DEXPERIENCE** platform **Editor** object instead of a V5 **Document** object.

V5 Automation

```
' Get the active Document
Set oDocument1 = CATIA.ActiveDocument

' Open a document using its Path and
' access the associated Document object
Set cDocuments = CATIA.Documents
Set oDocument2 = cDocuments.Open("xxx")
```

3DEXPERIENCE platform Automation

```
' Get the active Editor
Set oEditor1 = CATIA.ActiveEditor

' Open a PLMEntity object and
' access the associated Editor object
Set oPLMOpenService = CATIA.GetSessionService("PLMOpenService")
Set oEditor2 = oPLMOpenService.PLMOpen oPLMEntity
```

Accessing Objects Edited through the Document and the Editor

V5 **Document** objects are virtual objects that are specialized depending on their content, like for example the **PartDocument** or **DrawingDocument** objects. Most of the time a macro expects specific types of documents, the generic **Document** object can so be replaced by a more specific object, like a **DrawingDocument** object, that provides dedicated methods to access its specific content. This is illustrated in the following macro that displays the number of sheets of a set of drawings.

V5 Automation

```
' Macro will process DrawingDocument objects
Dim oDrawingDocument As DrawingDocument

' Loop on the files of a folder
Set oFolder = CATIA.FileSystem.GetFolder("E:\tmp")
```

3DEXPERIENCE platform Automation

```
' Use the SearchService object to retrieve a set of
' VPMRepresentationReference objects that contain drawing data
Dim oSearchService As SearchService
Set oSearchService = CATIA.GetSessionService("Search")
Dim oDBSearch As DatabaseSearch
Set oDBSearch = oSearchService.DatabaseSearch
Set oDBSearch = oSearchService.DatabaseSearch
' The type must be a Product Representation Reference type
```

```

For each oFile in oFolder.Files
  ' "PLMRepresentationDS" is available with the PLMProductDS customization.
  ' Use your own customization types.
  If ( Right(oFile.Path, 8) = ".CATDrawing" ) Then
    ' Open the file: a DrawingDocument object is expected
    Set oDrawingDocument = CATIA.Documents.Open(oFile.Path) ' Loop on found PLM entities
    For each oPLMEntity in oDBSearch.Results
      ' Retrieve the DrawingRoot object using a specific
      ' method of the DrawingDocument object
      Set oDrawingRoot = oDrawingDocument.DrawingRoot

      ' Display the number of sheets
      msgbox oDrawingRoot.Sheets.Count
    End If
  Next
  ' Open the entity: Editor object will have no specific type
  Set oEditor = CATIA.GetSessionService("PLMOpenService").PLMOpen(oPLMEntity);
  ' The editor has no specific type, but drawing data is expected, so
  ' the generic GetItem method is used to retrieve the DrawingRoot object
  Set oDrawingRoot = oEditor.GetItem("DrawingRoot")

  ' Display the number of sheets
  msgbox oDrawingRoot.Sheets.Count
Next

```

For details on each type of V5 **Document** objects, consult the *Details of modifications on V5 modelers* part of this *Adoption* section.

When V5 documents are already in session, the specialized type of the **Document** object allows to select a specific kind of document. Some objects may not be always present in the selected document. For example a **PartDocument** object will always contain a **Part** object but not always a **Layout2DRoot** object (2D Layout for 3D Design root object), so the type of the **Document** object is not always discriminative. In **3DEXPERIENCE** platform, representation references may contain different types of data, so this last method is always required. The following example describes how to process all **Layout2DRoot** objects in session.

V5 Automation

```

' Macro processes edited documents containing 2D Layout data
Dim oDocument As Document
' Loop on all documents in session
For each oDoc in CATIA.Documents
  ' Select only part documents
  If ( TypeName (oDoc) = "PartDocument" ) Then
    ' Check if Documents contains a 2DLayoutRoot
    Set oLayout2DRoot = NULL
    On error resume next
    Set oLayout2DRoot = oDoc.Part.GetItem("CATLayout2DRoot")
    On error goto 0
    ' Test whether a 2D Layout Root has been found
    If not IsNull(oLayout2DRoot) Then
      ...
    End If
  End If
Next

```

3DEXPERIENCE platform Automation

```

' Macro processes any editors editing 2D Layout data
Dim oEditor As Editor
' Loop on all editors in session
For each oEditor in CATIA.Editors
  ' Get The Active Object
  Set oObj = oEditor.ActiveObject
  ' Test whether it is a 2D Layout Root
  If ("Typename(oObj) = "Layout2DRoot") Then
    ...
  End If
Next

```

Changed Capabilities Related to Editing Objects in Session

Selection

The **Selection** object aggregation is moved from the V5 **Document** object to the **3DEXPERIENCE** platform **Editor** object:

V5 Automation

```

' Access Selection object
Set oSelection = CATIA.ActiveDocument.Selection

```

3DEXPERIENCE platform Automation

```

' Access Selection object
Set oSelection = CATIA.ActiveEditor.Selection

```

Cameras

V5 **Camera** objects are aggregated under the V5 **Document** object. **3DEXPERIENCE** platform **Camera** objects are only available in representations references in **3DEXPERIENCE** platform and should be accessed through a service on the **Editor** object: the **VisuServices** object.

V5 Automation

```

' Access the Cameras collection object
' from the Document object
Dim oPartDoc As PartDocument
Set oPartDoc = CATIA.ActiveDocument

Set oCameras = oPartDoc.Cameras

```

3DEXPERIENCE platform Automation

```

' Access the Cameras collection object from
' the VisuServices object
Dim oEditor As Editor
Set oEditor = CATIA.ActiveEditor

Dim oEditor As VisuServices
Set oEditor = Editor.GetService("VisuService")

Set oCameras = oService.Cameras

```

Commands and Workbenches

Some V5 interactive commands can be launched from a macro using the **StartCommand** method of the **Application** object. This capability still exists in **3DEXPERIENCE** platform but the name and behavior of the commands may have changed, so each usage of this method must be considered independently, based on **3DEXPERIENCE** platform commands interactive behavior.

The same applies for interactive workbenches that can be accessed through the **GetWorkbenchId** and **SetWorkbenchId** methods of the **Application** object.

The GetWorkbench Method

The **GetWorkbench** method of the **Document** object gives access, in V5, to a variety of objects, improperly called **Workbench** objects, that generally can't be seen as aggregated by another objects and offer access to edition contexts and creation or calculation services. Such V5 objects are not directly linked to the interactive notion of workbench described upper.

In **3DEXPERIENCE** platform such objects will be retrieved as **Service** objects either available at session level or at editor level. The list of V5 objects accessible through the **GetWorkbench** method is not listed here. For details on availability and name of the **3DEXPERIENCE** platform object equivalent to a V5 **Workbench** object, consult the *Details of modifications on V5 modelers* part of this *Adoption* section dedicated to the corresponding V5 modeler.

Extracting a Consistent Set of Objects from the Database

The **SendToService** object and its **CreateSendTo** factory method of the **Application** object are removed. Extracting a consistent set of data to send to a partner or a

supplier should now be interactively done thanks to Export-Import 3DXML with authoring commands. Refer to VPM Multi-Discipline collaboration product documentation for details.

Creating Objects

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open 3DEXPERIENCE platform PLM infrastructure. It explains how the V5 document centric model and more precisely the **Documents** collection object has been replaced to accommodate creation of persistent objects in database.

In V5, the **Document** object is the key object that allows persistency. Through the **Documents** collection object, new **Document** objects can be created as a repository for persistent data, modifications performed in session can be saved on disk, and existing **Document** objects loaded into session.

In 3DEXPERIENCE platform, persistent data is stored in the PLM server, either as a PLM object and its attributes and links, or as the streamed (black box) content of a specific PLM objects: the representation references.

This article focuses on object creation. Saving and retrieving objects is described in the [Bringing Objects into Session And Saving Objects](#) article.

Creating Objects

For 3DEXPERIENCE platform object that are stored in reference representations. The creation mode has not changed, but PLM objects should be created using the **PLMNewService** object. This service should be used in three steps:

1. Creating a template entity that will receive any required PLM attributes and their default values.
2. Modifying the template entity PLM attributes values if necessary.
3. Creating and editing actual PLM entities in session using the template entity.

```
V5 Automation                                3DEXPERIENCE platform Automation
' Get the PLMNewService
Dim oNewService As PLMNewService
' Creating a new Document object in memory Set oNewService = CATIA.GetSessionService("PLMNewService")
Dim newDoc As PartDocument
Set newDoc = CATIA.Documents.Add("Part")      ' Create And Edit entity
                                                Dim oEditor As Editor
                                                Set oEditor = oNewService.PLMCreate("3DShape")
```

Note: Drawing representations are created with default values for standards and sheet style. See the [Drafting Changes](#) article for more information.

Creating Objects as Copy of Existing Objects

3DEXPERIENCE platform is no more document based, so the **NewFrom** method of the **Documents** collection object has no meaning anymore. No equivalent capability is currently provided.

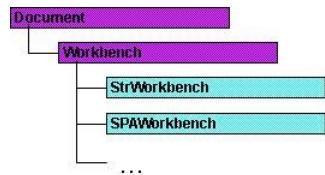
Accessing Objects Through Generic Methods

Technical Article

Abstract

This article explains how to transform your V5 automation macros to take advantage of the Open 3DEXPERIENCE platform PLM infrastructure. It explains how objects that were accessed in V5 through generic methods, meaning methods allowing to retrieve more than one type of object, can be retrieved in 3DEXPERIENCE platform.

In V5 a variety of objects could be seen as aggregated or not by a parent object depending on the products that were available in the installation, for example a **StrWorkbench** object could be obtained from a **ProductDocument** object if a product featuring structure design capability was installed and a **SPAWorkbench** object could be obtained if a product featuring space analysis capability was installed.



Some of those objects were retrieved using two generic methods that have disappeared in 3DEXPERIENCE platform, the **GetWorkbench** method of the **Document** object and the **GetTechnologicalObject** method of the **Product** object, and a text string that described what object was to be retrieved. For example, the **StrWorkbench** object was retrieved the following way:

```
V5 Automation
' Retrieving the StrWorkbench object associated with a Document using
'   - the GetWorkbench generic method
'   - a string identifying the object to be retrieved
Dim oStrWorkbench As StrWorkbench
Set oStrWorkbench = CATIA.ActiveDocument.GetWorkbench("StrWorkbench")
'
Set oNewDoc = CATIA.Documents.Add("Part")
```

The generic **GetItem** method of the **AnyObject** object still exists in 3DEXPERIENCE platform but may not allow anymore to retrieve all objects that were available in V5.

How to retrieve objects currently available in 3DEXPERIENCE platform is described in the 3DEXPERIENCE platform Adoption documents related to each object. This purpose of this article is to provide a way to retrieve the same information starting from the V5 generic method for the most common objects.

Objects	Methods	Strings	Retrieved Objects	Comments
---------	---------	---------	-------------------	----------

Product	GetTechnologicalObject	CATAnnotationSets	AnnotationSets	In 3DEXPERIENCE platform, Annotation Sets are only defined in a 3D Shape Representation Reference. Refer to 3D Functional Tolerancing & Annotation Changes .
ProcessDocument	GetItem	CATAnnotationSets	AnnotationSets	Use the PCBService object.
ProductDocument	GetWorkbench	PCBWorkbench	PCBWorkbench	Part of the objects aggregated by the SPAWorkbench object are now Service Objects . Refer to Space Analysis Changes .
ProductDocument	GetWorkbench	SPAWorkbench	SPAWorkbench	

InInterfaces

This table summarizes the V5 InInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to explain how to replace or get rid of them.

Changed Objects

Objects	Properties or Methods	Signatures	Comments
		Still available with the following property or method changes.	
Application	CreateSendTo	CreateSendTo (CATIASendToService oMail)	Refer to the following article: Accessing and Editing Objects in Session .
	Property ActiveDocument (Read)	ActiveDocument (CATIADocument oDocument)	Refer to the following article: Accessing and Editing Objects in Session .
	Property Documents (Read)	Documents (CATIADocuments oDocuments)	Refer to the following article: Accessing and Editing Objects in Session .
	Property FileSearchOrder (Read)	FileSearchOrder (inCATBSTR oDefaultFilePathBSTR)	Refer to the following article: Overview of 3DEXPERIENCE platform Adoption for Automation .
	Property FileSearchOrder (Write)	FileSearchOrder (CATBSTR iDefaultFilePathBSTR)	Refer to the following article: Overview of 3DEXPERIENCE platform Adoption for Automation .
CGRAdhesionSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CGRFormat setting repository. See the settings section in Automating Settings .	
Document		Not supported any longer. Refer to the following article: Accessing and Editing Objects in Session .	
DocumentationSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the GeneralHelp setting repository. See the settings section in Automating Settings .	
Documents		Not supported any longer. Refer to the following article: Accessing and Editing Objects in Session .	
GeneralSessionSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the GeneralGeneral setting repository. See the settings section in Automating Settings .	
PrintersSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATPrintersSettings setting repository. See the settings section in Automating Settings .	
SearchSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the GeneralFind setting repository. See the settings section in Automating Settings .	
SelectedElement	Properties or Methods	Signatures	Comments
	Property Document (Read)	Document (CATIADocument oDocument)	Refer to the following article: Accessing and Editing Objects in Session .
SelectionSets		This class or interface is out of the scope of the current release and is not currently supported.	
SendToService		Not supported any longer. Refer to the following article: Accessing and Editing Objects in Session .	
TreeVizManipSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the Tree setting repository. See the settings section in Automating Settings .	
VisualizationSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the VisualizationRepository setting repository. See the settings section in Automating Settings .	
VrmlSettingAtt		Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the VrmlFormat setting repository. See the settings section in Automating Settings .	

System

This table summarizes the V5 System Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.

- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments	
AccesslogStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsAudit setting repository. See the settings section in Automating Settings .	
CacheSettingAtt	Not supported any longer. Not supported any longer.	
CommandStatisticsSettingAtt	This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsCommand setting repository. See the settings section in Automating Settings .	
DisconnectionSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the Disconnection setting repository. See the settings section in Automating Settings .	
ErrorlogStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsErrorlog setting repository. See the settings section in Automating Settings .	
FileAccessStatisticsSettingAtt	Not supported any longer. Refer to the following article: Automating Settings .	
GeneralStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsGlobal setting repository. See the settings section in Automating Settings .	
GlobalStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsGlobal setting repository. See the settings section in Automating Settings .	
Properties or Methods	Signatures	Comments
LicenseSettingAtt	GetDemoModeInfo (inCATBSTR ioAdminLevel, inout CATBSTR ioLocked, out boolean oModified)	
	GetGrantedLicensesList (long iDefaultLicenses, CATSafeArrayVariant oTabDLName)	
	SetDemoModeLock (boolean iLock)	
	Property DemoMode (Read) (boolean oDemoMode)	
	Property DemoMode (Write) (boolean iDemoMode)	
	Not supported any longer.	
	This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATMemWarning setting repository. See the settings section in Automating Settings .	
	Not supported any longer.	
PCSStatisticsSettingAtt	This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsPCS setting repository. See the settings section in Automating Settings .	
ServerStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsActivity setting repository. See the settings section in Automating Settings .	
SessionStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsSession setting repository. See the settings section in Automating Settings .	
WorkbenchStatisticsSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATStatisticsWorkbench setting repository. See the settings section in Automating Settings .	

Product Changes

Technical Article

Abstract

This article explains how to transform your V5 automation macros to take advantage of the Open 3DEXPERIENCE platform PLM infrastructure. It focuses on how the API has been modified to comply to the new 3DEXPERIENCE platform product model.

The ProductDocument Object

In V5, the repository for Product objects and composition of Product objects is the product document. In 3DEXPERIENCE platform, this information is stored in the PLM server. Products are PLM objects that must be retrieved and brought in memory through by Search, Explore and Open operations like described in [Bringing Objects into Session And Saving Objects](#).

V5 Automation

```
' Get the Search Service
Dim oSearchService As SearchService
```

3DEXPERIENCE platform Automation

```

Set oSearchService = CATIA.GetSessionService("Search")
' Instantiate a DatabaseSearch object

Dim oDBSearch As DatabaseSearch
Set oDBSearch = oSearchService.DatabaseSearch

' A Product document is known by its path
sIdentifier="E:/myDocument.CATProduct"

' Its content is loaded through the Documents collection
Set oProductDocument = CATIA.Documents.Open("sIdentifier")

' Retrieve the listed entities
Dim cPLMEntities As PLMEntities
Set cPLMEntities = oDBSearch.Results

' Get the PLMOpen service & Open the first retrieved entity
Set oPLMOpenService = CATIA.GetSessionService("PLMOpenService")
Set oEditor = oPLMOpenService.PLMOpen(cPLMEntities.Item(1))

```

To access product objects in memory, a specific service has been provided on the Editor object: the **PLMProductService** service object. Here is a typical example of how to retrieve the root product of a product structure.

V5 Automation

```

' Retrieving the root Product object of the edited product document
Dim oProduct As Product
Set oProduct = oProductDocument.Product

' Retrieving the collection of root entities edited in the editor
Dim cPLMEntities2 As PLMEntities
Set cPLMEntities2 = oPLMProductService.EditedContent

' Retrieving the root product from the Editor
' as the first edited PLMEntity
Dim oVPMReference As VPMReference
Set oVPMReference = cPLMEntities2.Item(1)

```

3DEXPERIENCE platform Automation

```

' Retrieving the PLMProductService object as a service associated w:
Dim oPLMProductService As PLMProductService
Set oPLMProductService = oEditor.GetService("PLMProductService")

' Retrieving the collection of root entities edited in the editor
Dim cPLMEntities As PLMEntities
Set cPLMEntities = oPLMProductService.EditedContent

' Retrieving the root product from the Editor
' as the first edited PLMEntity
Dim oVPMReference As VPMReference
Set oVPMReference = cPLMEntities.Item(1)

```

The Product Object and the Products Collection

Instance and References

The V5 **Product** object, that represents both concepts of instance and reference of a product, is split in 3DEXPERIENCE platform into a **VPMReference** object, retrieved in the previous sample, and a **VPMInstance** object. The collection of **Product** objects that allowed to retrieved the instances aggregated by a product reference is so replaced in 3DEXPERIENCE platform by a collection of **VPMInstance** objects that contains the position information and points on its reference:

V5 Automation

```

' Loop on instances of the previously retrieved product
Dim oPosition As Position
Dim oReferenceOfInstance As Product

Dim oInstance As Product
For each oInstance In oProduct.Products
  ' Retrieve the position of the instance
  Set oPosition = oInstance.Position

  ' Retrieve the reference of this instance
  Set oReferenceOfInstance = oInstance.ReferenceProduct
Next

```

3DEXPERIENCE platform Automation

```

' Loop on instances of the previously retrieved product
Dim oPosition As Position
Dim oReferenceOfInstance As VPMReference

Dim oInstance As VPMInstance
For each oInstance In oVPMReference.Instances
  ' Retrieve the position of the instance
  Set oPosition = oInstance.Position

  ' Retrieve the reference of this instance
  Set oReferenceOfInstance = oInstance.ReferenceInstanceOf
Next

```

The **Position** object allows you to retrieve and modify the position of the instance.

Representations

As the V5 **Product** object, the 3DEXPERIENCE platform **VPMReference** object has representations that are instances of representation object, represented by the **VPMRepInstance** and **VPMRepReference** objects.

V5 Automation

```

' Retrieve the Part representation of the previously retrieved product
Dim oReferenceOfInstance As Product

Dim oPartDocument As PartDocument
Set oPartDocument = oReferenceOfInstance.GetMasterShapeRepresentation(False)

```

3DEXPERIENCE platform Automation

```

' Loop on representations of the previously retrieved product
Dim oReferenceOfRepInstance As VPMReference

Dim oRepInstance As VPMRepInstance
For each oRepInstance In oReferenceOfInstance.RepInstances
  ' Retrieve the reference of this instance
  Set oReferenceOfRepInstance = oRepInstance.ReferenceInstances
Next

```

User Properties

Properties that can be associated to a V5 Product reference are in 3DEXPERIENCE platform attributes of the **PLMEntity** objects, parent class of all PLM objects. In V5 properties can be dynamically added to any reference while in 3DEXPERIENCE platform attributes must be declared, for a specific type of product, during the customization stage of the installation of the PLM server. Their values can then be accessed or modified:

V5 Automation

```

' Acces the supplier property of a product reference
Dim oReference As Product
Set oReference = ...

' Properties are accessed as a collection of Parameter objects
Dim oProperties As Parameters
Set oProperties = oReference.Parameters
Dim oProperty As Parameter
Set oProperty = oParameters.Item("V_Supplier")
sStringSupplier = oProperty.ValueAsString

```

3DEXPERIENCE platform Automation

```

' Acces the supplier attribute of a product reference
Dim oReference As VPMReference
Set oReference = ...

' Attributes are accessed through the GetAttributeValue
' method inherited from the PLMEntity object
sStringSupplier = oReference.GetAttributeValue("V_Supplier")

```

Publications and Connection

There is currently no access to the equivalent **3DEXPERIENCE** platform functionality.

References

There is currently no access to the equivalent **3DEXPERIENCE** platform functionality.

Access to Knowledgeware Parameters and Relations

In V5 the **Product** object offers two collections to access parameters and relations. In **3DEXPERIENCE** platform, the **KnowledgeServices** service object should be retrieved from the **Editor** object.

The GetTechnologicalObject Method

The V5 `GetTechnologicalObject` method provides access to a variety of objects linked to a product. For details on availability and name of the **3DEXPERIENCE** platform object equivalent to such a V5 object, consult either the [Accessing Objects Through Generic Methods](#) article or the *Details of modifications on V5 modelers* part of this *Adoption* section dedicated to the V5 modeler corresponding to the object.

ProductStructureInterfaces

This table summarizes the V5 ProductStructureInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
Analyze	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .
AssemblyConvertor	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .
Product	Not supported any longer. Refer to the following article: Product Changes .
ProductDocument	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .
Products	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .
Publication	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .
Publications	Not supported any longer. Refer to the following article: 3DEXPERIENCE platform Adoption Patterns .

Knowledge Advisor Changes

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM 2.0 infrastructure. It focuses on Knowledge Advisor.

Two V5 Knowledge Advisor Automation objects do not exist any longer with **3DEXPERIENCE** platform:

- The **FeatureGenerator** object

This object is exposed in V5, but without any implementation. So it cannot be used in V5, should not be found in V5 macros, and is removed in **3DEXPERIENCE** platform.

- The **Loop** object

As a specialization of the **KnowledgeActivateObject** object, it does not provide any properties or methods in V5. If you used it in V5, you can replace it with this object.

The setting objects are replaced with the generic one and XML files. Refer to [KnowledgeInterfaces](#) changed objects for details, and to [Automating Settings](#) to know how to use the generic setting object.

KnowledgeInterfaces

This table summarizes the V5 KnowledgeInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Moved Objects

This table summarizes the V5 KnowledgeInterfaces objects that still exist in **3DEXPERIENCE** platform, but that are located in other frameworks. You can use this table to get their typelib.

Objects	V5 Typelibs	3DEXPERIENCE platform Frameworks	3DEXPERIENCE platform Typelibs
ConstraintSatisfaction	KnowledgewareTypeLib OptimizationIDLIfc		OptimizationTLB

FreeParameter	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB
FreeParameters	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB
Optimization	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB
OptimizationConstraint	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB
OptimizationConstraints	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB
Optimizations	KnowledgewareTypeLib OptimizationIDLItf	OptimizationTLB

Changed Objects

Objects	Comments					
	Properties or Methods	Signatures	Comments			
			Frameworks	Typelibs	Objects	Methods
DesignTable	Property FilePath (Read)	FilePath (inCATBSTR oFilePath)	Not supported any longer. 1 replacing method	KnowledgeIDLItf	KnowledgewareTypeLib	DesignTable SheetRepRef
	Property FilePath (Write)	FilePath (CATBSTR iFilePath)	Not supported any longer. 1 replacing method	KnowledgeIDLItf	KnowledgewareTypeLib	DesignTable SheetRepRef
FeatureGenerator			Not supported any longer. Refer to the following article: Knowledge Advisor Changes .			
KnowledgeSheetSettingAtt			Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the Knowledge setting repository. See the settings section in Automating Settings .			
LanguageSheetSettingAtt			Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the KnowledgeLanguage setting repository. See the settings section in Automating Settings .			
Loop			Not supported any longer. Refer to the following article: Knowledge Advisor Changes .			
			Still available in another framework: KnowledgeIDLItf and from the KnowledgewareTypeLib typelib. Some properties or methods are changed:			
	Properties or Methods	Signatures	Comments			
Relations	CreateDesignTable	CreateDesignTable (CATBSTR iName, in CATBSTR iComment, in boolean iCopyMode, in CATBSTR iSheetPath, CATIADesignTable oDesignTable)	Not supported any longer. 1 replacing method	KnowledgeIDLItf	KnowledgewareTypeLib	Relations CreateDesignTab
	CreateHorizontalDesignTable	CreateHorizontalDesignTable (CATBSTR iName, in CATBSTR iComment, in boolean iCopyMode, in CATBSTR iSheetPath, CATIADesignTable oDesignTable)	Not supported any longer. 1 replacing method	KnowledgeIDLItf	KnowledgewareTypeLib	Relations CreateHorizontal
	Property Optimizations (Read)	Optimizations (inCATIAOptimizations oOptimizations)	Not supported any longer. 1 replacing method	KnowledgeIDLItf	KnowledgewareTypeLib	KnowledgeObjects GetKno
ReportGenerationSheetSettingAtt			Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the Report setting repository. See the settings section in Automating Settings .			
ToleranceSheetSettingAtt			Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CkeTolerance setting repository. See the settings section in Automating Settings .			
UnitsSheetSettingAtt			Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the NumberDisplay setting repository. See the settings section in Automating Settings .			

CATIAV4Interfaces

This table summarizes the V5 CATIAV4Interfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to.
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
InteropSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the Interoperability setting repository.

	See the settings section in Automating Settings .
MigrBatchSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the SpecV4 setting repository.
SpecV4SettingAtt	See the settings section in Automating Settings .
V4MasterModel	This class or interface is out of the scope of the current release and is not currently supported. Not supported any longer.
V4V5SpaceSettingAtt	This is a setting interface, replaced with the generic one: SettingRepository . Use it with the V4V5Space setting repository. See the settings section in Automating Settings .
V4WritingSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the V4Writing setting repository. See the settings section in Automating Settings .

CATIdSettingsInterfaces

This table summarizes the V5 CATIdSettingsInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
IgesSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
StepSettingAtt	Use it with the IGES setting repository. See the settings section in Automating Settings .

CATSdeSettingInterfaces

This table summarizes the V5 CATSdeSettingInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
StepSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
	Use it with the STEP setting repository. See the settings section in Automating Settings .

Mechanical Modeler and Part Applications Changes

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM infrastructure. It focuses on how the Part object and objects related to Part Applications can be retrieved.

The PartDocument Object

The **PartDocument** object is in V5 the key object to access Part objects. In **3DEXPERIENCE** platform, those objects persist in the 3DShape representation of the Product modeler. In the following example, such a representation, whose type always derives from **VPMRepReference**, is retrieved through a set of attributes. Any kind of representation can be retrieved this way. No predefined attribute shows that the retrieved object is actually a 3DShape. Generally, company processes allow to identify 3DShapes by defining standard values for specific attributes. Content can also be tested using the **GetItem** method of the retrieved **PLMEntity** object.

V5 Automation	3DEXPERIENCE platform Automation
' The PartDocument is known by its path sPart = "myPart.CATPart"	' Use the SearchService object to retrieve a VPMRepresentationReference ' from its identification attributes Dim oSearchService As SearchService Set oSearchService = CATIA.GetSessionService("Search") Dim oDBSearch As DatabaseSearch Set oDBSearch = oSearchService.DatabaseSearch oDBSearch.BaseType = "VPMRepReference" oDBSearch.AddEasyCriteria "PLM_ExternalID", "myPart" oSearchService.Search
' Open the file: a Part Document Dim oPartDocument As PartDocument Set oPartDocument = CATIA.Documents.Open(sPart)	' Get the first found PLM entity (other versions may exist) Set oEntity = oDBSearch.Results.Item(1)
' Retrieve the Part object Set oPart = oPartDocument.Part	' Open the entity: the Editor object has no specific type Set oEditor = CATIA.GetSessionService("PLMOpenService").PLMOpen(oEntity)
	' The editor has no specific type, but part data is expected, so to ' retrieve the Part object one can use:

```
' - either the ActiveObject property (illustrated here)
' - or the generic GetItem method (illustrated in next example)
Set oPart = oEditor.ActiveObject
```

More information on how to replace usage of the Document object is described in [Accessing and Editing Objects in Session](#).

The Product property of the PartDocument Object

In V5, a Product object is contained in each Part document. In **3DEXPERIENCE** platform, this is not true anymore. V5 scenarios retrieving the Product object contained in the Part Document generally aimed at identifying which product is a Part when exploring a product structure, to process data inside the Part document:

V5 Automation	3DEXPERIENCE platform Automation
<pre>Dim oProduct As Product Set oProduct = ... ' Loop on instances attached to this reference Dim oChildProduct As Product For each oChildProduct In oProduct.Products ' Retrieve the reference of the considered instance Dim oRefOfInstance As Product Set oRefOfInstance = oChildProduct.ReferenceProduct ' Identify which product of a structure is a Part: If (typename (oReference.Parent) = "PartDocument") Then ' Use the PartDocument to retrieve the Part msgbox oReference.Parent.Part.Name End If Next</pre>	<pre>Dim oVPMReference As VPMReference Set oVPMReference = ... ' Loop on instances attached to this reference Dim oInstance As VPMInstance For each oInstance In oVPMReference.Instances ' Retrieve the reference of the considered instance Dim oRefOfInstace As VPMReference Set oRefOfInstace = oInstance.ReferenceInstanceOf ' Retrieve first associated representation Set oRepInstance = oRefOfInstace.RepInstances.Item(1) ' Retrieve as the e RepReference of the RepInstance Set oRefOfRepInstance = oRepInstance.ReferenceInstanceOf ' Identify if it is a 3DShape: see above If (...) Then ' Use VPMRepReference to retrieve the Part msgbox oRefOfRepInstance.GetItem("Part").Name End If Next</pre>

Note that in V5, references and instances have the same Product type, while in **3DEXPERIENCE** platform, they have distinct types: VPMReference and VPMInstance.

Objects Related to Part Applications

Retrieving a Part object is described above. Objects that are retrieved from the Part object in V5, like Sketch objects, Shape objects, Body objects, HybridShape objects, or Layout2DRoot objects are retrieved the same way in **3DEXPERIENCE** platform:

V5 Automation	3DEXPERIENCE platform Automation
<pre>' Retrieve the main Body object Set oBody = oPart.MainBody ' Retrieve the 2D Layout Root Set o2DLayoutRoot = oPart.GetItem("CATLayoutRoot")</pre>	<pre>' Retrieve the main Body object Set oBody = oPart.MainBody ' Retrieve the 2D Layout Root Set o2DLayoutRoot = oPart.GetItem("CATLayoutRoot")</pre>

MecModInterfaces

This table summarizes the V5 MecModInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to.
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Moved Objects

This table summarizes the V5 MecModInterfaces objects that still exist in **3DEXPERIENCE** platform, but that are located in other frameworks. You can use this table to get their typelib.

Objects V5 Typelibs	3DEXPERIENCE platform Frameworks	3DEXPERIENCE platform Typelibs
Part	MECMOD	CATMmrAutomationInterfaces
		CATOpnsSectionIDL

Changed Objects

	Objects	Comments			
		Properties or Methods	Signatures	Comments	
FixTogether					This class or interface is out of the scope of the current release and is not currently supported.
FixTogethers					This class or interface is out of the scope of the current release and is not currently supported.
					Still available in another framework: CATMmrAutomationInterfaces and from the MECMOD typelib.
					Some properties or methods are changed:
InstanceFactory			BeginInstanceFactory (CATBSTR iNameOfReference, in CATBSTR iNameOfDocument)		Not supported any longer. I replacing method
				Frameworks	Typelibs
				CATMmrAutomationInterfaces	MECMOD
					InstanceFactory
					BeginInstanceFactory
PartDocument					Not supported any longer. Refer to the following article: Mechanical Modeler and Part Applications Changes .
					Not supported any longer.
PartInfrastructureSettingAtt					This is a setting interface, replaced with the generic one: SettingRepository . Use it with the 3DShapeInfrastructure3DShape setting repository. See the settings section in Automating Settings .

CATAssemblyInterfaces

This table summarizes the V5 CATAssemblyInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and

that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to.
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
AsmConstraintSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the AssemblyDesign setting repository. See the settings section in Automating Settings .
AsmGeneralSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the AssemblyDesign setting repository. See the settings section in Automating Settings .
AssemblyBoolean	This class or interface is out of the scope of the current release and is not currently supported.
AssemblyFeature	This class or interface is out of the scope of the current release and is not currently supported.
AssemblyFeatures	This class or interface is out of the scope of the current release and is not currently supported.
AssemblyHole	This class or interface is out of the scope of the current release and is not currently supported.
AssemblyPocket	This class or interface is out of the scope of the current release and is not currently supported.
AssemblySplit	This class or interface is out of the scope of the current release and is not currently supported.

Drafting Changes

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM 2.0 infrastructure. It focuses on how Drawing objects can be accessed.

Opening a DrawingDocument Object

The **DrawingDocument** object is the V5 key object to access Drawing objects. In **3DEXPERIENCE** platform, those objects persist in the Drawing representation of the Product modeler. In the following example, such a representation, whose type will always derive from **VPMRepReference**, is retrieved through a set of attributes. Any kind of representation can be retrieved this way. No predefined attribute shows that the retrieved object is a Drawing. Generally, company processes allow Drawings to be identified by defining standard values for specific attributes. Content can also be tested using the **GetItem** method of the retrieved **PLMEntity** object.

V5 Automation

```
' The Drawing document is known by its path
sDrawing = "myDrawing.CATDrawing"

' Open the file: a Drawing document
Dim oDrawingDocument As DrawingDocument
Set oDrawingDocument = CATIA.Documents.Open(sDrawing)

' Retrieve the DrawingRoot object
Dim oDrawingRoot As DrawingRoot
Set oDrawingRoot = oDrawingDocument.DrawingRoot
```

3DEXPERIENCE platform Automation

```
' Use a SearchService object to retrieve a VPMRepReference
' object from its identification attributes
Dim oSearchService As SearchService
Set oSearchService = CATIA.GetSessionService("Search")
Dim oDBSearch As DatabaseSearch
Set oDBSearch = oSearchService.DatabaseSearch
oDBSearch.BaseType = "VPMRepReference"
' To retrieve the Drawing by its Title, use V_Name attribute
' To retrieve the Drawing by its Name, use PLM_ExternalID attribute
oDBSearch.AddEasyCriteria "V_Name", "myDrawing"
oSearchService.Search

' Get, for example, the first found PLM entity
Dim oEntity As PLMEntity
Set oEntity = oDBSearch.Results.Item(1)
' Open the entity: the Editor object has no specific type
Dim oEditor As Editor
CATIA.GetSessionService("PLMOpenService").PLMOpen oEntity,oEditor
' Retrieve the DrawingRoot object
Dim oDrawingRoot As DrawingRoot
Set oDrawingRoot = oEditor.ActiveObject
```

If the Drawing representation is retrieved from navigating a product structure in session, its content can be directly accessed using the **GetItem** method:

V5 Automation

```
' ---- Not a V5 scenario ----
Dim oVPMReference As VPMReference
Set oVPMReference = ...
' Loop on instances attached to this reference
For each oInstance In oVPMReference.Instances
  ' Retrieve the reference of the considered instance
  Dim oRefOfInstance As VPMReference
  Set oRefOfInstance = oInstance.ReferenceInstanceOf
  ' Retrieve the first associated representation
  Dim oRepInstance As VPMRepInstance
  Set oRepInstance = oRefOfInstance.RepInstances.Item(1)
  ' Retrieve the RepReference of the Repinstance
  Dim oRefOfRepInstance As VPMRepReference
  Set oRefOfRepInstance = oRepInstance.ReferenceInstanceOf
  ' Identify if it is a Drawing representation: see above
  Dim oDrawingRoot As DrawingRoot
  If (...) Then
    ' Use the VPMRepReference object to retrieve the Drawing Root reference
    Set oDrawingRoot = oRefOfRepInstance.GetItem("CATDrawingAccess")
  End If
Next
```

3DEXPERIENCE platform Automation

```
Dim oVPMReference As VPMReference
Set oVPMReference = ...
' Loop on instances attached to this reference
For each oInstance In oVPMReference.Instances
  ' Retrieve the reference of the considered instance
  Dim oRefOfInstance As VPMReference
  Set oRefOfInstance = oInstance.ReferenceInstanceOf
  ' Retrieve the first associated representation
  Dim oRepInstance As VPMRepInstance
  Set oRepInstance = oRefOfInstance.RepInstances.Item(1)
  ' Retrieve the RepReference of the Repinstance
  Dim oRefOfRepInstance As VPMRepReference
  Set oRefOfRepInstance = oRepInstance.ReferenceInstanceOf
  ' Identify if it is a Drawing representation: see above
  Dim oDrawingRoot As DrawingRoot
  If (...) Then
    ' Use the VPMRepReference object to retrieve the Drawing Root reference
    Set oDrawingRoot = oRefOfRepInstance.GetItem("CATDrawingAccess")
  End If
Next
```

More information on how to replace the usage of the V5 **Document** object is described in [Accessing and Editing Objects in Session](#).

Creating a DrawingDocument Object

In V5, Drawing documents are created through the collection of documents. In **3DEXPERIENCE** platform, a **PLMNewService** object should be used as explained in the [Creating Objects](#) article. For drawing representations, the standard and Sheet style are set with a default value when creating the representation, and if required can be re-set after creation:

V5 Automation	3DEXPERIENCE platform Automation <pre>' Get the PLMNewService Dim oNewService As PLMNewService Set oNewService = CATIA.GetSessionService("PLMNewService") ' Create a drawing representation Dim oEditor As Editor oNewService.PLMCreate "Drawing", oEditor ' Create a new DrawingDocument object in memory Dim newDoc As DrawingDocument Set newDoc = CATIA.Documents.Add("Drawing") ' Get the drawing root from the Editor Dim oDrawingRoot As DrawingRoot Set oDrawingRoot = oEditor.ActiveObject ' Set the drawing standard oDrawingRoot.Standard = "myStandard" ' Set the sheet style to the automatically created sheet oDrawingRoot.ActiveSheet.SheetStyle = "mySheetStyle"</pre>
----------------------	---

Setting the Drawing Standard

In V5, the drawing standard is represented by values of an enum, allowing to set only a predefined set of standards. In **3DEXPERIENCE** platform, corporate standards can be defined and the VB APIs allows to reference such standards by name. The type of the **3DEXPERIENCE** platform Standard property is so now a string and the **DrawingService** service allows you to get the list of available standards.

V5 Automation	3DEXPERIENCE platform Automation <pre>Dim oDrawingRoot As DrawingRoot Set oDrawingRoot = ... ' V5 way ' Possible standard values are described by an enum Dim enumStd As catDrawingStandard enumStd = catISO Dim iNbStd As Integer iNbStd = myDrawingService.NumberOfStandards ReDim arrayOfStdNames(iNbStd - 1) oDrawingService.StandardNamesList arrayOfStdNames stdName = arrayOfStdNames(1) ' Set the standard to the Drawing myDrwRoot.Standard = stdName</pre>
----------------------	--

Setting the Drawing Sheet Style

In V5, a sheet style can be set using interactive commands but Automation only allows you to set values of each part of the style individually. In **3DEXPERIENCE** platform, it is still possible to individually set each part of a sheet style, like the paper orientation, but it is also possible to globally set predefined or customer defined styles associated with a standard.

V5 Automation	3DEXPERIENCE platform Automation <pre>Dim oDrawingSheet As DrawingSheet Set oDrawingSheet = ... ' V5 way ' Set paper orientation using the PageSetup object linked to the Sheet oDrawingSheet.PageSetup.Orientation catPaperLandscape oDrawingSheet.SheetStyle = stringStyle ' Set paper size using a property of the sheet oDrawingSheet.PaperSize = catPaperA4 ' Overriding values is still possible through the Sheet object properties oDrawingSheet.PaperSize = catPaperA4</pre>
----------------------	--

Creating Interactive Views

New objects and methods to create interactive views are proposed to replace no more accessible methods of the **GenerativeViewBehavior** object.

Creating a Front View

V5 Automation	3DEXPERIENCE platform Automation <pre>' Get a DrawingViews collection Dim cViews As DrawingViews Set cViews = ... ' Create the DrawingView object first Dim oFrontView As DrawingView Set oFrontView = cViews.Add("Front View") ' Get a DrawingViewGenerativeBehavior object</pre>
----------------------	--

```

' from the DrawingView object
' even if the View is not generative
Dim oViewGenB As DrawingViewGenerativeBehavior
Set oViewGenB = oFrontView.GenerativeBehavior
' View Anchor point
' View Anchor point
Dim xPos, yPos As double
xPos = 10.
yPos = 10.

' Vectors defining the view projection plane
Dim ix1, iy1, iz1, ix2, iy2, iz2 As double
ix1 = 1.
iy1 = 0.
iz1 = 0.
ix2 = 0.
iy2 = 1.
iz2 = 0.

' Define it a a front view
' Directly create the front view with all required data
Dim oFrontView As DrawingView
Set oFrontView = cViews.AddFrontView( _
    ix, iy, "MyViewName", _
    ix1, iy1, iz1, ix2, iy2, iz2)

```

Creating a Projection View

The same way, the `AddProjectionView` method of the **DrawingViews** collection object should be used.

Creating an Isometric View

The same way, the `AddIsometricView` method of the **DrawingViews** collection object should be used.

Accessing the View Projection Plane

In V5, methods of the **GenerativeViewBehavior** object allow you to retrieve or set the view projection plane. In **3DEXPERIENCE** platform, they are available with the **DrawingView** object.

In V5, methods of the **GenerativeViewLinks** object allow you to add a link or to retrieve the 3D link from the Generative View. In **3DEXPERIENCE** platform, they are available with the **DrawingGenView** object.

So, both methods to retrieve **GenerativeViewLinks** (`get_GenerativeLinks`) or **GenerativeViewBehavior** (`get_GenerativeBehavior`) from the **DrawingView** are useless.

V5 Automation

```

' Get a DrawingView object
Dim oView As DrawingView
Set oView = ...
' Get a GenerativeViewBehavior object
' from the DrawingView object
Dim oViewGenB As DrawingViewGenerativeBehavior
Set oViewGenB = oView.GenerativeBehavior
' Get the View Projection Plane
Dim ix1, iy1, iz1, ix2, iy2, iz2 As double
oViewGenB.GetProjectionPlane( _
    ix1, iy1, iz1, ix2, iy2, iz2)
' Set the View Projection Plane
oViewGenB.SetProjectionPlane( _
    ix1, iy1, iz1, ix2, iy2, iz2)

```

3DEXPERIENCE platform Automation

```

' Get a DrawingView object
Dim oView As DrawingView
Set oView = ...
' No need of the GenerativeViewBehavior object
oView.GetProjectionPlane( _
    ix1, iy1, iz1, ix2, iy2, iz2)
' Set the View Projection Plane
oView.SetProjectionPlane( _
    ix1, iy1, iz1, ix2, iy2, iz2)

```

Creating, Managing and Modifying a Generative View

In V5 generative views are accessed through the **GenerativeViewBehavior** object. In **3DEXPERIENCE** platform, this object is replaced with several objects. The different usage scenarios are described below.

Creating Generative Views

The **DrawingDefineGenView** object contains all methods to create generative views. It can be retrieved from the **3DEXPERIENCE** platform **DrawingDefineGenView** property of the **DrawingViews** object. The following examples show the creation of a generative view in V5 and in **3DEXPERIENCE** platform.

Managing Generative View Properties

The **DrawingGenViewProperties** object should be used. It can be retrieved from the **3DEXPERIENCE** platform **DrawingGenViewProp** property if the **DrawingService** object. Default values are initialized from generative view settings.

Modifying the Generative View

The **DrawingGenView** object should be used. This object proposes methods to add a breakout, create a break view, etc. It can be retrieved from the **3DEXPERIENCE** platform **DrawingGenView** property of the **DrawingView** object.

Example

The following example shows how to create a generative front view where hidden lines are drawn.

V5 Automation

```

' Create a view
Dim cViews As DrawingViews
Set cViews = ...
Dim oView as DrawingView
Set oView = cViews.Add("FrontView")
' Get a DrawingViweGenerativeBehavior object
' from the DrawingView object

```

3DEXPERIENCE platform Automation

```

' Create a view
Dim cViews As DrawingViews
Set cViews = ...
Dim oView as DrawingView
' Get the DefineGenView object from
' the DrawingViews collection object

```

```

Dim oViewGenB As DrawingViewGenerativeBehavior      Dim oDefGenView As DrawingDefineGenView
Set oViewGenB = oView.GenerativeBehavior           Set oDefGenView = cViews.DrawingDefineGenView
                                                 ' Retrieve the drawing generative service object from
                                                 ' the CATIA service
Dim oDrwGenServ As DrawingGenService             Set oDrwGenServ =
                                                 CATIA.ActiveEditor.GetService("CATDrawingGenService")

                                                 ' Retrieve a Generative View Properties
Dim oGenViewProp As DrawingGenViewProperties       Set oGenViewProp = oDrwGenServ.DrawingGenViewProp
                                                 ' Activate the hidden line mode
oGenViewProp.HiddenLineMode = catHRLModeOn        oGenViewProp.HiddenLineMode = catHRLModeOn
                                                 ' Get the 3D to draw:

                                                 ' - To draw the complete product structure get a VPMRootOccurrence
                                                 '   from the root product.
                                                 '   For example, suppose the window containing the 3D is the first
                                                 '   opened window in the CATIA.
Dim o3DPLMToDraw As VPMRootOccurrence           Set o3DPLMToDraw = CATIA.Editors.Item(1).ActiveObject
Set o3DPLMToDraw = CATIA.Editors.Item(1).ActiveObject

                                                 ' OR ...

                                                 ' - To draw a sub part number numInst of the product structure
                                                 '   get a VPMInstance to identify the sub product.
Dim oPLMRootOcc As VPMRootOccurrence           Set oPLMRootOcc = CATIA.Editors.Item(1).ActiveObject
Set oPLMRef As VPMReference                   Set oPLMRef = oPLMRootOcc.ReferenceRootOccurrenceOf
Set o3DPLMToDraw As VPMInstance                Dim o3DPLMToDraw = oPLMRef.Instances.item(numInst)

                                                 ' Assign it as an item of a list
Dim myListofPLMInst(0) As CATSafeArrayVariant     myListofPLMInst(0) = o3DPLMToDraw
myListofPLMInst(0) = o3DPLMToDraw
                                                 ' Define the View anchor point
Dim xPos, yPos As Double                         xPos = 10.
yPos = 10.

                                                 ' Define the View projection plane
Dim myProjPlane(5)                             myProjPlane(0) = 1
myProjPlane(1) = 0
myProjPlane(2) = 0
myProjPlane(3) = 0
myProjPlane(4) = 1
myProjPlane(5) = 0

                                                 ' Define this view as a front view
Set oView = oDefGenView.DefineFrontView xPos, yPos,
myListofPLMInst, myProjPlane, "", True, oGenViewProp

' Define the View anchor point
oViewGenB.x = 10.
oViewGenB.y = 10.

' Define this view as a front view, with the
' XY plane (in oPartToDraw) as projection plane
oViewGenB.DefineFrontView 1, _
0, -
0, -
0, -
1, -
0
                                                 ' Define the View projection plane
Dim myProjPlane(5)
myProjPlane(0) = 1
myProjPlane(1) = 0
myProjPlane(2) = 0
myProjPlane(3) = 0
myProjPlane(4) = 1
myProjPlane(5) = 0

                                                 ' Define this view as a front view
Set oView = oDefGenView.DefineFrontView xPos, yPos,
myListofPLMInst, myProjPlane, "", True, oGenViewProp

' Define the View anchor point
oViewGenB.Document = oPartToDraw

```

Mapping Table between the V5 GenerativeViewLinks Object and the 3DEXPERIENCE platform DrawingGenView Object

V5 IDL interfaces	Methods/Properties	3DEXPERIENCE platform IDL interfaces	Methods/Properties
GenerativeViewLinks AddLink	DrawingGenView	PutLinks (Write)	
GenerativeViewLinks FirstLink	DrawingGenView	GetLink (Read)	
GenerativeViewLinks NextLink	DrawingGenView	GetLink (Read)	
GenerativeViewLinks RemoveAllLinks	DrawingGenView	RemoveLink (Write)	
GenerativeViewLinks CopyLinksTo	DrawingGenView	PutLinks (R/W)	

Mapping Table between the V5 GenerativeViewBehavior Object and 3DEXPERIENCE platform Objects

V5 IDL interfaces	Methods/Properties	3DEXPERIENCE platform IDL interfaces	Methods/Properties
GenerativeViewBehavior Document (Read)	DrawingGenView	GetAssociatedRootProduct	
GenerativeViewBehavior Document (Write)	Not Replaced		
GenerativeViewBehavior SetProjectionPlane	DrawingGenView	ModifyViewPlane	
GenerativeViewBehavior GetProjectionPlane	DrawingView	GetProjectionPlane	
GenerativeViewBehavior GetProjectionPlaneNormal	Not replaced		
GenerativeViewBehavior DefineFrontView	DrawingDefineGenView	DefineFrontView	
GenerativeViewBehavior DefineIsometricView	DrawingDefineGenView	DefineIsometricView	
GenerativeViewBehavior DefineProjectionView	DrawingDefineGenView	DefineProjectionView	
GenerativeViewBehavior DefineSectionView	DrawingDefineGenView	DefineSectionView	
GenerativeViewBehavior DefineCircularDetailView	DrawingDefineGenView	DefineCircularDetailView	
GenerativeViewBehavior DefinePolygonalDetailView	DrawingDefineGenView	DefinePolygonalDetailView	
GenerativeViewBehavior DefineAuxiliaryView	DrawingDefineGenView	DefineAuxiliaryView	
GenerativeViewBehavior DefineUnfoldedView	DrawingDefineGenView	DefineUnfoldedView	
GenerativeViewBehavior DefineStandAloneSection	DrawingDefineGenView	DefineStandAloneSection	
GenerativeViewBehavior DefineTPSSectionView	Not Replaced		
GenerativeViewBehavior DefineBrokenView	DrawingGenView	AddBrokenView	
GenerativeViewBehavior DefineBox3DView	DrawingGenView	AddClippingBox	
GenerativeViewBehavior UnBreak	DrawingGenView	UnBreak	
GenerativeViewBehavior UnClip	DrawingGenView	UnClip	
GenerativeViewBehavior UnBreakout	DrawingGenView	UnBreakout	
GenerativeViewBehavior UnBreakoutNum	Not replaced		
GenerativeViewBehavior ApplyBreakoutTo	DrawingGenView	ApplyBreakoutTo	
GenerativeViewBehavior DefineBreakout	DrawingGenView	AddBreakout	
GenerativeViewBehavior DefineCircularClippingView	DrawingGenView	AddClippingWithCircle	

GenerativeViewBehavior DefineCircularExactClipping	DrawingGenView	AddClippingWithCircle
GenerativeViewBehavior DefinePolygonalClippingView	DrawingGenView	AddClippingWithProfile
GenerativeViewBehavior DefinePolygonalExactClipping	DrawingGenView	AddClippingWithProfile
GenerativeViewBehavior Update	DrawingGenView	Update
GenerativeViewBehavior ForceUpdate	DrawingGenView	ForceUpdate
GenerativeViewBehavior get_ParentView	DrawingGenView	get_ParentView
GenerativeViewBehavior SetAxisSysteme	DrawingGenView	SetAxisSysteme
GenerativeViewBehavior GetAxisSysteme	DrawingGenView	GetAxisSysteme
GenerativeViewBehavior ColorInheritanceMode	DrawingGenViewProperties	ColorInheritanceMode
GenerativeViewBehavior PointsProjectionMode	DrawingGenViewProperties	PointsProjectionMode
GenerativeViewBehavior PointsSymbol	DrawingGenViewProperties	PointsSymbol
GenerativeViewBehavior RepresentationMode	DrawingGenViewProperties	RepresentationMode
Not replaced.		
GenerativeViewBehavior SetGPSName	This information is mentioned during the view creation	
Not replaced.		
GenerativeViewBehavior GetGPSName	This information is mentioned during the view creation	
GenerativeViewBehavior ImageViewMode	DrawingGenViewProperties	ImageViewMode
GenerativeViewBehavior LimitBoundingBox	DrawingGenViewProperties	LimitBoundingBox
GenerativeViewBehavior HiddenLineMode	DrawingGenViewProperties	HiddenLineMode
GenerativeViewBehavior FilletRepresentation	DrawingGenViewProperties	FilletRepresentation

DraftingInterfaces

This table summarizes the V5 DraftingInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to.
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Moved Objects

This table summarizes the V5 DraftingInterfaces objects that still exist in **3DEXPERIENCE** platform, but that are located in other frameworks. You can use this table to get their typelib.

Objects	V5 Typelibs	3DEXPERIENCE platform Frameworks	3DEXPERIENCE platform Typelibs
DrawingArrow	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingArrows	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingDimExtLine	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingDimLine	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingDimValue	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingDimension	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingDimensions	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingLeader	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingLeaders	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingTable	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingTables	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingText	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingTextProperties	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingTextRange	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingTexts	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingWelding	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	
DrawingWeldings	DRAFTINGITF CATAnnotationIDLItf	CATANNOTITF	

Changed Objects

Objects	Comments									
DraftingSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the DraftingOptions setting repository. See the settings section in Automating Settings .									
DrawingDocument	Not supported any longer. Refer to the following article: Drafting Changes . Still available in another framework: DraftingIDLItf and from the DRAFTINGITF typelib. Some properties or methods are changed: <table border="1"> <thead> <tr> <th>Properties or Methods</th> <th>Signatures</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>Property Standard (Read)</td> <td>Standard (CatDrawingStandard oStandard)</td> <td>Refer to the following article: Drafting Changes.</td> </tr> <tr> <td>Property Standard (Write)</td> <td>Standard (CatDrawingStandard iStandard)</td> <td>Refer to the following article: Drafting Changes.</td> </tr> </tbody> </table>	Properties or Methods	Signatures	Comments	Property Standard (Read)	Standard (CatDrawingStandard oStandard)	Refer to the following article: Drafting Changes .	Property Standard (Write)	Standard (CatDrawingStandard iStandard)	Refer to the following article: Drafting Changes .
Properties or Methods	Signatures	Comments								
Property Standard (Read)	Standard (CatDrawingStandard oStandard)	Refer to the following article: Drafting Changes .								
Property Standard (Write)	Standard (CatDrawingStandard iStandard)	Refer to the following article: Drafting Changes .								
DrawingRoot	Still available in another framework: DraftingIDLItf and from the DRAFTINGITF typelib. Some properties or methods are changed: <table border="1"> <thead> <tr> <th>Properties or Methods</th> <th>Signatures</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>Property Standard (Read)</td> <td>Standard (CatDrawingStandard oStandard)</td> <td>Refer to the following article: Drafting Changes.</td> </tr> <tr> <td>Property Standard (Write)</td> <td>Standard (CatDrawingStandard iStandard)</td> <td>Refer to the following article: Drafting Changes.</td> </tr> </tbody> </table>	Properties or Methods	Signatures	Comments	Property Standard (Read)	Standard (CatDrawingStandard oStandard)	Refer to the following article: Drafting Changes .	Property Standard (Write)	Standard (CatDrawingStandard iStandard)	Refer to the following article: Drafting Changes .
Properties or Methods	Signatures	Comments								
Property Standard (Read)	Standard (CatDrawingStandard oStandard)	Refer to the following article: Drafting Changes .								
Property Standard (Write)	Standard (CatDrawingStandard iStandard)	Refer to the following article: Drafting Changes .								
DrawingView	Still available in another framework: DraftingIDLItf and from the DRAFTINGITF typelib. Some properties or methods are changed: <table border="1"> <thead> <tr> <th>Properties or Methods</th> <th>Signatures</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>Property GenerativeBehavior (Read)</td> <td>GenerativeBehavior (CATIGenerativeViewBehavior oGenerativeBehavior)</td> <td>Refer to the following article: Drafting.</td> </tr> <tr> <td>Property GenerativeLinks (Read)</td> <td>GenerativeLinks (CATIGenerativeViewLinks oGenerativeLinks)</td> <td>Refer to the following article: Drafting.</td> </tr> </tbody> </table>	Properties or Methods	Signatures	Comments	Property GenerativeBehavior (Read)	GenerativeBehavior (CATIGenerativeViewBehavior oGenerativeBehavior)	Refer to the following article: Drafting .	Property GenerativeLinks (Read)	GenerativeLinks (CATIGenerativeViewLinks oGenerativeLinks)	Refer to the following article: Drafting .
Properties or Methods	Signatures	Comments								
Property GenerativeBehavior (Read)	GenerativeBehavior (CATIGenerativeViewBehavior oGenerativeBehavior)	Refer to the following article: Drafting .								
Property GenerativeLinks (Read)	GenerativeLinks (CATIGenerativeViewLinks oGenerativeLinks)	Refer to the following article: Drafting .								

Properties or Methods	Signatures	Comments
ApplyBreakoutTo	ApplyBreakoutTo (CATIAGenerativeViewBehavior iParentView)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects M DraftingIDLIf DRAFTINGITF DrawingGenView Apply
DefineAuxiliaryView	DefineAuxiliaryView (double iXStartPoint, in double iYStartPoint, in double iXEndPoint, in double YEndPoint, in short iSideToDraw, in CATIAGenerativeViewBehavior iParentViewGenerativeBehavior)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineBox3DView	DefineBox3DView (CATIABase iBoxableObject)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects M DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DefineBreakout	DefineBreakout (CATSafeArrayVariant iProfil, in CATSafeArrayVariant iPlane1, in CATSafeArrayVariant iPlane2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects M DraftingIDLIf DRAFTINGITF DrawingGenView AddB
DefineBrokenView	DefineBrokenView (CATSafeArrayVariant iBrokenLinesExtremities, in double iXDirection, in double iYDirection)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects M DraftingIDLIf DRAFTINGITF DrawingGenView AddB
DefineCircularClippingView	DefineCircularClippingView (double XCenter, in double YCenter, in double Radius)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DefineCircularDetailView	DefineCircularDetailView (double iXCenter, in double iYCenter, in double iRadius, in CATIAGenerativeViewBehavior iParentViewGenerativeBehavior)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineCircularExactClippingView	DefineCircularExactClippingView (double XCenter, in double YCenter, in double Radius)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DrawingViewGenerativeBehavior	DefineFrontView (double iX1, in double iY1, in double iZ1, in double iX2, in double iY2, in double iZ2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineFrontView	DefineIsometricView (double iX1, in double iY1, in double iZ1, in double iX2, in double iY2, in double iZ2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineIsometricView	DefineIsometricView (double iX1, in double iY1, in double iZ1, in double iX2, in double iY2, in double iZ2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DefinePolygonalClippingView	DefinePolygonalClippingView (CATSafeArrayVariant profil)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DefinePolygonalDetailView	DefinePolygonalDetailView (CATSafeArrayVariant iProfile, in CATIAGenerativeViewBehavior iParentViewGenerativeBehavior)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefinePolygonalExactClippingView	DefinePolygonalExactClippingView (CATSafeArrayVariant profil)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingGenView AddC
DefineProjectionView	DefineProjectionView (CATIAGenerativeViewBehavior iParentViewGenerativeBehavior, in CatProjViewType iType)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineSectionView	DefineSectionView (CATSafeArrayVariant iProfile, in CATBSTR iSectionType, in CATBSTR iProfileType, in short iSideToDraw, in CATIAGenerativeViewBehavior iParentViewGenerativeBehavior)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIf DRAFTINGITF DrawingDefineGenView
DefineStandAloneSection	DefineStandAloneSection (CATSafeArrayVariant profil, in CATBSTR type_of_section, in CATBSTR type_of_profile,	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects

		in CATSafeArrayVariant iPlane, in short iSide) DefineTPSSectionView (CATSafeArrayVariant iProfile, in CATBSTR iSectionType, in CATBSTR iProfileType, in short iSideToDraw, in CATIAGenerativeViewBehavior iParentViewGenerativeBehavior) DefineUnfoldedView (double iX1, in double iY1, in double iZ1, in double iX2, in double iY2, in double iZ2)	DraftingIDLIfc DRAFTINGITF DrawingDefineGenView Refer to the following article: Drafting Changes .
DefineUnfoldedView			Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIfc DRAFTINGITF DrawingDefineGenView
ForceUpdate		ForceUpdate()	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView Force
GetAxisSysteme		GetAxisSysteme (CATIABase oProduct, out CATIABase oAxisSysteme)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView GetAx
GetGPSName		GetGPSName (inCATBSTR ioGPSName)	Refer to the following article: Drafting Changes .
GetProjectionPlane		GetProjectionPlane (double oX1, out double oY1, out double oZ1, out double oX2, out double oY2, out double oZ2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingView GetProj
GetProjectionPlaneNormal		GetProjectionPlaneNormal (double oXNormal, out double oYNormal, out double oZNormal)	Refer to the following article: Drafting Changes .
SetAxisSysteme		SetAxisSysteme (CATIABase iProduct, in CATIABase iAxisSysteme)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView SetAx
SetGPSName		SetGPSName (CATBSTR iGPSName)	Refer to the following article: Drafting Changes .
SetProjectionPlane		SetProjectionPlane (double iX1, in double iY1, in double iZ1, in double iX2, in double iY2, in double iZ2)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIfc DRAFTINGITF DrawingGenView Modif
UnBreak		UnBreak()	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView UnBre
UnBreakout		UnBreakout()	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView UnBre
UnBreakoutNum		UnBreakoutNum (short iBreakoutNumber)	Refer to the following article: Drafting Changes .
UnClip		UnClip()	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView UnCli
Update		Update()	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects Meth DraftingIDLIfc DRAFTINGITF DrawingGenView Updat
Property ColorInheritanceMode (Read)		ColorInheritanceMode (Cat3DColorInheritanceMode oMode)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIfc DRAFTINGITF DrawingGenView Proper
Property Document (Read)		Document (CATIABase oDocument)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIfc DRAFTINGITF DrawingGenView GetAs
Property FilletRepresentation (Read)		FilletRepresentation (CatFilletRepresentation oRep)	Not supported any longer. 1 replacing method Frameworks TypeLibs Objects DraftingIDLIfc DRAFTINGITF DrawingGenView Proper
			Not supported any longer.

			1 replacing method	
		Frameworks	TypeLibs	Objects
	Property HiddenLineMode (Read)	HiddenLineMode (CatHiddenLineMode oMode)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
	Property ImageViewMode (Read)	ImageViewMode (CatImageViewMode oMode)		Not supported any longer.
	Property LimitBoundingBox (Read)	LimitBoundingBox (double oMode)		1 replacing method
	Property ParentView (Read)	ParentView (CATIADrawingView oParentView)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
	Property PointsProjectionMode (Read)	PointsProjectionMode (CatPointsProjectionMode oMode)		Not supported any longer.
	Property PointsSymbol (Read)	PointsSymbol (short oSymbol)		1 replacing method
	Property RepresentationMode (Read)	RepresentationMode (CatRepresentationMode oMode)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
	Property ColorInheritanceMode (Write)	ColorInheritanceMode (Cat3DCColorInheritanceMode iMode)		Not supported any longer.
	Property Document (Write)	Document (CATIABase iDocument)		1 replacing method
	Property FilletRepresentation (Write)	FilletRepresentation (CatFilletRepresentation iRep)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
	Property HiddenLineMode (Write)	HiddenLineMode (CatHiddenLineMode iMode)		Refer to the following article: Drafting Changes .
	Property ImageViewMode (Write)	ImageViewMode (CatImageViewMode iMode)		Not supported any longer.
	Property LimitBoundingBox (Write)	LimitBoundingBox (double iMode)		1 replacing method
	Property PointsProjectionMode (Write)	PointsProjectionMode (CatPointsProjectionMode iMode)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
	Property PointsSymbol (Write)	PointsSymbol (short iSymbol)		Not supported any longer.
	Property RepresentationMode (Write)	RepresentationMode (CatRepresentationMode iMode)	DraftingIDLIf	DRAFTINGITF DrawingGenViewProper
DrawingViewGenerativeLinks		Not supported any longer. 1 replacing object.		
		Frameworks	TypeLibs	Objects
		DraftingIDLIf	DRAFTINGITF	DrawingGenView

Drafting2DInterfaces

This table summarizes the V5 Drafting2DInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
Layout2DSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
Use it with the CAT2DLToolsOptions setting repository. See the settings section in Automating Settings .	

3D Functional Tolerancing & Annotation Changes

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open 3DEXPERIENCE platform PLM2.0 infrastructure. It focuses on how 3D Functional Tolerancing & Annotation objects can be accessed and created.

Annotation Sets

In V5, 3D Functional Tolerancing & Annotations objects are obtained through the **AnnotationSets** collection object, retrieved:

- In a CATPart document

Directly under the **Part** object:

```
Set cAnnotationSets = oPart.AnnotationSets
```

- In a CATProduct document:

Through the **GetTechnologicalObject** method of a **Product** object:

```
Set cAnnotationSets = oProductDocument.Product.GetTechnologicalObject("CATAnnotationSets")
```

- In a CATProcess document:

Through the **GetItem** method of a **ProcessDocument** object:

```
Set cAnnotationSets = oProcessDocument.GetItem("CATAnnotationSets")
```

In 3DEXPERIENCE platform, annotation Sets is only defined in a 3D Shape representation reference. The **AnnotationSets** collection object is retrieved by using the **AnnotationSets** property of the **Part** object.

Hereunder the script create a dimension of a hole. The surface of the hole is selected before launching the macro.

V5 Automation

```
' A Part Document is the active document in the session
Dim oPartDocument As PartDocument
Set oPartDocument = CATIA.Documents.ActiveDocument

' Get the Part object
Dim oPart As Part
Set oPart = oPartDocument.Part

' Get the Selection
Dim oSel As Selection
Set oSel = oPartDocument.Selection

V5 and 3DEXPERIENCE platform Automation
' Get the AnnotationSet from the part feature
Dim oAnnotSet As AnnotationSet
Set oAnnotSet = oPart.AnnotationSets.Item(1)

' Get the collection of UserSurfaces
Dim cUserSurfs As Object
Set cUserSurfs = oPart.UserSurfaces

' Get the Annotation Factory
Dim oAnnotFact As AnnotationFactory
Set oAnnotFact = oAnnotSet.AnnotationFactory

' The dimensioned surface must be selected
If (oSel.Count = 1) Then
    ' Create the hole dimension
    Dim myRef1 As Reference
    Set myRef1 = mySel.Item(1).Reference

    Dim myUserSurf1 As UserSurface
    Set myUserSurf1 = cUserSurfs.Generate(myRef1)

    Dim myDim3DForHole As Annotation
    Set myDim3DForHole =
        oAnnotFact.CreateSemanticDimension(myUserSurf1, 1, 1)
End If
```

3DEXPERIENCE platform Automation

```
' A Representation is opened in the session.
' The Part object can be retrieved
' from the active editor.

' Get the Part object
Dim oPart As Part
Set oPart = CATIA.ActiveEditor.ActiveObject

' Get the Selection
Dim oSel As Selection
Set oSel = CATIA.ActiveEditor.Selection
```

V5 and 3DEXPERIENCE platform Automation

```
' Get the AnnotationSet from the part feature
Dim oAnnotSet As AnnotationSet
Set oAnnotSet = oPart.AnnotationSets.Item(1)

' Get the collection of UserSurfaces
Dim cUserSurfs As Object
Set cUserSurfs = oPart.UserSurfaces

' Get the Annotation Factory
Dim oAnnotFact As AnnotationFactory
Set oAnnotFact = oAnnotSet.AnnotationFactory

' The dimensioned surface must be selected
If (oSel.Count = 1) Then
    ' Create the hole dimension
    Dim myRef1 As Reference
    Set myRef1 = mySel.Item(1).Reference

    Dim myUserSurf1 As UserSurface
    Set myUserSurf1 = cUserSurfs.Generate(myRef1)

    Dim myDim3DForHole As Annotation
    Set myDim3DForHole =
        oAnnotFact.CreateSemanticDimension(myUserSurf1, 1, 1)
End If
```

Below is the script to create an **AnnotationSet** object from the **Part** object.

V5 Automation

```
' A Part Document is the active document in the session
Dim oPartDocument As PartDocument
Set oPartDocument = CATIA.Documents.ActiveDocument

' Get the Part object
Dim oPart As Part
Set oPart = oPartDocument.Part
```

3DEXPERIENCE platform Automation

```
' The Part object can be retrieved
' from the active editor.

' Get the Part object
Dim oPart As Part
Set oPart = CATIA.ActiveEditor.ActiveObject
```

V5 and 3DEXPERIENCE platform Automation

```

' Get the Annotation Set collection from the Part feature
Dim cAnnotSets As AnnotationSets
Set cAnnotSets = oPart.AnnotationSets

' Create an Annotation Set
Dim oAnnotSet As AnnotationSet
Set oAnnotSet = cAnnotSets.Add("ISO_3D")

```

Note: Only one **AnnotationSet** object can be aggregated by the **Part** object.

CATTPSInterfaces

This table summarizes the V5 CATTPSInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Properties or Methods		Signatures	Comments
AnnotationSetTransformIntoAssemblySet	Transform		Transform (CATBSTR iAssemblyannotationSetName)	This method is out of the scope of the current release and is not currently supported. Still available in another framework: CATTPSIDLItf and from the AnnotationTypeLib typelib. Some properties or methods are changed:
AnnotationSets	AddInAProduct		AddInAProduct (CATIAProduct iProduct, in CATBSTR iStandard, CATIAAnnotationSet oSet)	Refer to the following article: 3D Functional Tolerancing & Annotation Changes .
DMUToSettingAtt				Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATTPSEditor setting repository. See the settings section in Automating Settings .
FTAInfraSettingAtt				Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATTPSEditor setting repository. See the settings section in Automating Settings .
FTASettingAtt				Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the CATTPSEditor setting repository. See the settings section in Automating Settings . Still available in another framework: CATTPSIDLItf and from the AnnotationTypeLib typelib. Some properties or methods are changed:
UserSurface	AddReferenceInAProductCtx		AddReferenceInAProductCtx (CATIAProduct iProdInst, in CATIAReference iSupport)	Refer to the following article: 3D Functional Tolerancing & Annotation Changes .
UserSurfaces	GenerateInAProductCtx		GenerateInAProductCtx (CATIAProduct iProduct, in CATIAProduct iProdInst, in CATIAReference iSupport, CATIAUserSurface oUserSurf)	Refer to the following article: 3D Functional Tolerancing & Annotation Changes .

CATSmInterfaces

This table summarizes the V5 CATSmInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
CatalogSHMObjectSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
ViewCharacteristicCurvesSettingAtt	Use it with the AerospaceSheetmetalDesign setting repository. See the settings section in Automating Settings .
	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .

Use it with the [AerospaceSheetmetalDesign](#) setting repository.
See the settings section in [Automating Settings](#).

Space Analysis Changes

Technical Article

Abstract

This article explains how to transform your V5 automation macros to take advantage of the Open 3DEXPERIENCE platform PLM2.0 infrastructure. It focuses on how macros using V5 measures and sections should be modified to accommodate the new 3DEXPERIENCE platform object model.

Working with Sections

Part of the functionality accessible by the V5 Section object is available in the current release. **Section** objects were retrieved in V5:

- either through the **SPAWorkbench** object that aggregated a collection of all section objects of the active document
- or through the **GetItem** method applied to the root **Product** object of a Product document, that allowed to retrieve the same collection.

In 3DEXPERIENCE platform, the **SectionService** objects behaves like a collection and allows to retrieve any **Section** object visible in the editor on which the **SectionService** object has been requested.

V5 Automation	3DEXPERIENCE platform Automation
' Retrieve the Sections collection from the active document Dim cSections As Sections Set cSections = _ CATIA.ActiveDocument.Product.GetTechnologicalObject("Sections")	' Retrieve the SectionService from the active editor Dim oSectionService As SectionService Set oSectionService = _ CATIA.ActiveEditor.GetService("SectionService")
' Retrieve the first Section object Dim oSection As Section Set oSection = cSections.Item(1)	' Retrieve the first Section object Dim oSection As Section Set oSection = oSectionService.Item(1)

On the Section object itself, the **GetPosition** and the **SetPosition** methods remain the same, the **Export** method now creates curve in a 3DShape representation identified by its **Part** object, instead of a **PartDocument** object in V5, and the **CutMode** property has been renamed into a **SceneRender** property.

Working with Measures

The **Measurable** object allowed to analyze dimensions related to an objects or between two objects. A **Measurable** object was obtained from an object through **GetMeasurable** method of the **SPAWorkbench** object. Depending on the measured object, methods of the **Measurable** object worked or not. For example the **GetDirection** method worked when the underlying object could be seen as a line and failed otherwise

In 3DEXPERIENCE platform, a Measurable object from a predefined type can be retrieved through the **GetMeasurable** method of the **MeasurableService** object. For example a **MeasurableLine** object has to be requested on an object in order to use the **GetDirection** method. If the underlying object cannot be seen as a line, the **GetMeasurable** method will fail.

V5 Automation	3DEXPERIENCE platform Automation
' Retrieve the SPA workbench Dim oSPAWorkbench As SPAWorkbench Set oSPAWorkbench = _ CATIA.ActiveDocument.GetWorkbench("SPAWorkbench")	' Retrieve the MeasurableService from the active editor Dim oMeasurableService As MeasurableService Set oMeasurableService = _ CATIA.ActiveEditor.GetService("MeasurableService")
' Get a Reference object from a Line object Dim oReferenceOnLine As Reference Set oReferenceOnLine = part1.CreateReferenceFromObject(oMyLine)	' Get a MeasurableLine object from a Line object Dim oMeasurableLine As MeasurableLine Set oMeasurableLine = _ oMeasurableService.GetMeasurable(oMyLine, CAAMeasurableLine)
' Get a Measurable object on this line through the reference Dim oMeasurable As Measurable Set oMeasurable = oSPAWorkbench.GetMeasurable(oReferenceOnLine)	' Get the direction of the line as an array of coordinates Dim aDirection (2) Set oMeasurableLine.GetDirection aDirection
' Get the direction of the line as an array of coordinates Dim aDirection (2) Set oMeasurable.GetDirection aDirection	' Get the direction of the line as an array of coordinates Dim aDirection (2) Set oMeasurableLine.GetDirection aDirection

The **GeometryName** property allowed, by returning a **CATMeasurableName** enumerated value, to retrieved how to the underlying object was seen by the Measurable object: **CatMeasurableLine** for lines, **CatMeasurablePlane** for planes and so on.

In 3DEXPERIENCE platform equivalent values have been provided as arguments of the **GetMeasurable** method: **CAAMeasurableLine** for lines, **CAAMeasurablePlane** for planes and so on. The measure type is so now requested by the programmer and not anymore deduced by the API, the property has so been removed. This has three consequences:

1. There is no equivalent to the **CatMeasurableUnknown** V5 value that was returned when underlying object was not measurable. If an object does not support the requested measure, the **GetMeasurable** method fails
2. **GetGeometryName** cannot be used anymore in 3DEXPERIENCE platform to determine how the object will be seen by the Measurable object. This is however still possible by requesting alternatively all possible measures on the object and processing failures
3. Methods available in V5 on the Measurable object have been dispatched in 3DEXPERIENCE platform in the different types of measurable objects. The names have sometimes been slightly changed to homogeneity purpose.

The following array give a correspondence between the V5 methods of the Measurable object and the available 3DEXPERIENCE platform methods.

V5 Objects	V5 Methods or Properties	Applied to GeometryName	3DEXPERIENCE platform Methods	3DEXPERIENCE platform Objects
GeometryName	any	None	None	
GetAngleBetween	any	AngleTo	CATIA.MeasurableBetween	
GetMinimumDistance	any	DistanceMinTo	CATIA.MeasurableBetween	
GetMinimumDistancePoints	any			
GetAngle	CatMeasurableCircle CatMeasurableCone	GetAngle	CATIA.MeasurableCircle CATIA.MeasurableCone	
Area	catMeasurableSurface CatMeasurableVolume CatMeasurableAxisSystem	GetArea	CATIA.MeasurableSurface CATIA.MeasurableVolume CATIA.MeasurableAxisSystem	

	CatMeasurableCircle	GetAxis	CATIA Measurable Circle
	CatMeasurableCone		CATIA Measurable Cone
	catMeasurableCylinder		CATIA Measurable Cylinder
	CatMeasurableCircle	GetCenter	CATIA Measurable Circle
	CatMeasurableSphere		CATIA Measurable Sphere
	catMeasurableSurface		CATIA Measurable Surface
	CatMeasurableVolume	GetCOfG	CATIA Measurable Volume
	CatMeasurableLine	GetDirection	CATIA Measurable Line
	CatMeasurableCurve	GetLength	CATIA Measurable Curve
	CatMeasurableLine	GetOrigin	CATIA Measurable Line
CATIA Measurable	Perimeter	catMeasurableSurface	GetPerimeter
	GetPlane	CatMeasurablePlane	GetPlane
		CatMeasurableCone	
	GetPoint	catMeasurableCylinder	GetPoint
		CatMeasurablePoint	
	GetPointsOnCurve	CatMeasurableCurve	
		CatMeasurableCone	GetPoints
	GetPointsOnAxis	catMeasurableCylinder	
		CatMeasurableCircle	
	Radius	catMeasurableCylinder	GetRadius
		CatMeasurableSphere	
	Volume	CatMeasurableVolume	GetVolume

Working with Distances.

The V5 **Distance** object and its associated collection, allowed to calculate distances between **Product** objects or groups of Product. Only the first one is available in **3DEXPERIENCE** platform. It has been extended to any measurable object and has been provided thought a specific type of measurable: the **MeasurableBetween** object.

The V5 **Distances** collection was aggregated by the **SPAWorkbench** object. See upper how to retrieve a measurable object in **3DEXPERIENCE** platform through the **MeasurableService** object.

The table below shows which V5 methods are supported in the current **3DEXPERIENCE** platform release.

V5 Object	V5 Methods or Properties	3DEXPERIENCE platform Methods	3DEXPERIENCE platform Objects	Comments
CATIADistance	ComputationType	none	none	Only the functionality equivalent to <code>CatDistanceMeasureType::catDistanceMeasureTypeMinimum</code> value is supported
	MethodType	none	none	Only the functionality equivalent to <code>CatDistanceComputationMode::catDistanceComputationTypeInsideOne</code> value is supported
	MinimumDistance	DistanceMinToPoint	CATIA Measurable Between	
	Compute	none	none	Now useless. In 3DEXPERIENCE platform computation is launched at each method call.
	All other Methods	none	none	Not Supported yet

SpaceAnalysisInterfaces

This table summarizes the V5 SpaceAnalysisInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
Clash	This class or interface is out of the scope of the current release and is not currently supported.
ClashResult	This class or interface is out of the scope of the current release and is not currently supported.
ClashResults	This class or interface is out of the scope of the current release and is not currently supported.
Clashes	This class or interface is out of the scope of the current release and is not currently supported.
Conflict	This class or interface is out of the scope of the current release and is not currently supported.
Conflicts	This class or interface is out of the scope of the current release and is not currently supported.
Distance	Not supported any longer. Refer to the following article: Space Analysis Changes .
Distances	Not supported any longer. Refer to the following article: Space Analysis Changes .
Inertia	This class or interface is out of the scope of the current release and is not currently supported.
Inertias	This class or interface is out of the scope of the current release and is not currently supported.
Measurable	Not supported any longer. Refer to the following article: Space Analysis Changes .
MeasureSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository . Use it with the MeasureSettings setting repository. See the settings section in Automating Settings .
SPAWorkbench	Not supported any longer. Refer to the following article: Space Analysis Changes . Still available in another framework: CATOpsSectionIDLItf and from the CATOpsSectionIDL typelib. Some properties or methods are changed:

	Properties or Methods	Signatures	Comments
	Export	Export (CATIAObject oDocument)	Not supported any longer. Refer to the following article: Space Analysis Changes . 1 replacing method
Section	IsEmpty	IsEmpty (long oIndicator)	This method is out of the scope of the current release and is not currently supported.
	Property AnnotatedViews (Read)	AnnotatedViews (CATIAAnnotatedViews oAnnotatedViews)	This method is out of the scope of the current release and is not currently supported.
	Property Behavior (Read)	Behavior (CatSectionBehavior oBehavior)	This method is out of the scope of the current release and is not currently supported.
	Property CutMode (Read)	CutMode (long oMode)	This method is out of the scope of the current release and is not currently supported.
	Property Group (Read)	Group (CATAGroup oGroup)	This method is out of the scope of the current release and is not currently supported.
	Property Height (Read)	Height (double oHeight)	This method is out of the scope of the current release and is not currently supported.
	Property Marker3Ds (Read)	Marker3Ds (CATIAMarker3Ds oMarker3Ds)	This method is out of the scope of the current release and is not currently supported.
	Property Thickness (Read)	Thickness (double oThickness)	This method is out of the scope of the current release and is not currently supported.
	Property Type (Read)	Type (CatSectionType oType)	This method is out of the scope of the current release and is not currently supported.
	Property Width (Read)	Width (double oWidth)	This method is out of the scope of the current release and is not currently supported.
	Property Behavior (Write)	Behavior (CatSectionBehavior iBehavior)	This method is out of the scope of the current release and is not currently supported.
	Property CutMode (Write)	CutMode (long iMode)	This method is out of the scope of the current release and is not currently supported.
	Property Group (Write)	Group (CATAGroup iGroup)	This method is out of the scope of the current release and is not currently supported.
	Property Height (Write)	Height (double iHeight)	This method is out of the scope of the current release and is not currently supported.
	Property Thickness (Write)	Thickness (double iThickness)	This method is out of the scope of the current release and is not currently supported.
	Property Type (Write)	Type (CatSectionType iType)	This method is out of the scope of the current release and is not currently supported.
	Property Width (Write)	Width (double iWidth)	This method is out of the scope of the current release and is not currently supported.
SectioningSettingAtt	Not supported any longer.		
	This is a setting interface, replaced with the generic one: SettingRepository .		
	Use it with the SectioningSettings setting repository. See the settings section in Automating Settings .		
Sections	Not supported any longer.		
	Refer to the following article: Space Analysis Changes .		

KinematicsInterfaces

This table summarizes the V5 KinematicsInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to.
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

	Objects	Comments
Dressup		This class or interface is out of the scope of the current release and is not currently supported.
Dressups		This class or interface is out of the scope of the current release and is not currently supported.
Joint	Not supported any longer.	
	1 replacing object.	
Joints	Not supported any longer.	
	1 replacing object.	
KinematicsWorkbench	Not supported any longer.	
	1 replacing object.	
Mechanism	Not supported any longer.	
	1 replacing object.	
MechanismCommand	Not supported any longer.	
	1 replacing object.	
	Not supported any longer.	

MechanismCommands	Frameworks	Typelibs	Objects
	CATKinMechanismIDLIf	CATKinMechanismIDL	KinCommands

Mechanisms

Business Process Knowledge Template Changes

Technical Article

Abstract

This article explains how to transform your V5 Automation macros to take advantage of the Open **3DEXPERIENCE** platform PLM2.0 infrastructure. It focuses on Business Process Knowledge Template.

V5 Business Process Knowledge Template is changed to Automated Design Process in **3DEXPERIENCE** platform. Since the objects handled are operations, the objects are renamed as follows:

V5 Objects 3DEXPERIENCE platform Objects

Behavior	DPCOperation
BehaviorExtension	Not supported any longer
BehaviorVBScript	DPCOperationVBScript
Behaviors	DPCOperations

The **Behavior** objects are accessed in V5 from the object they extend using the **BehaviorExtension** object retrieved thanks to calling the `GetItem` method against this extended object. The **BehaviorExtension** object aggregates a **Behavior** object that aggregates a **Behaviors** collection object where the Behavior objects associated to the extended object can be used.

This model changes with **3DEXPERIENCE** platform. The **DPCOperations** collection object can directly be retrieved from the extended object still thanks to the `GetItem` method. As a consequence:

- The **Behavior** object property **Behaviors** that returned the child **Behaviors** collection object is removed from the **DPCOperation** object.
- The **BehaviorExtension** object is removed. It has one property and one method. Use the **DPCOperations** collection object to replace it, as follows:

o `ExtensionClass` property
Replace a call to this property, such as `oBehaviorExtension.ExtensionClass`, with

```
Dim cOperations As DPCOperations
...
Dim sExtension As String
Set sExtension = cOperations.Parent.Name
```

o `SelectBehavior` method
Replace a call to this method, such as `oBehaviorExtension.SelectBehavior("BehaviorName")`, with

```
Dim cOperations As DPCOperations
...
Dim oOperation As DPCOperation
Set oOperation = cOperations.Operate("BehaviorName")
```

If the **BehaviorExtension** object is used in a script launched by a **BehaviorVBScript** operation object (behavior) to get the operation itself, use in **3DEXPERIENCE** platform the first argument of the macro, which is the operation, as the argument of the `Operate` method.

Refer to [DPCOperations Collection Object](#) to retrieve such an object.

In addition:

- The methods `Activate` and `Deactivate` of the **Behavior** object are removed. The best way to control the activity of an operation is to act on its "Activity" parameter. This can be done either through an EKL operation or through a VB launched by the VBScript Operation:
 1. EKL: Compute the state of the targeted operation and directly validate its "Activity" parameter.
 2. VB: Compute the state of the targeted operation and validate an output variable ("ActivateOrNot" for example) of the VBScript Operation. The "Activity" parameter of the targeted operation must be linked to "ActivateOrNot".
- The methods `Start`, `Done`, `Suspend`, and `Cancel` of the **BehaviorVBScript** object are removed from the **DPCOperationVBScript** object. They have empty implementations in V5, and can safely be removed from your macros in **3DEXPERIENCE** platform, if you still call them.

In addition to object renaming and model changes, some properties or methods of the **Behavior** object are mapped to properties and methods of the **DPCOperationVBScript** object. Refer to [CATBehaviorInterfaces](#) changed objects for details.

CATBehaviorInterfaces

This table summarizes the V5 CATBehaviorInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments		
Still available in another framework: CATBehaviorIDLIf and from the DPCOperationsTypeLib typelib. Some properties or methods are changed:			
Properties or Methods	Signatures		
Activate	Activate()		
Deactivate	Deactivate()		
GetInput	GetInput (CATBSTR pName,		
	1 replacing method		
	Not supported any longer.		
Frameworks	Typelibs	Objects	Methods

		CATIABase oValue)	CATBehaviorIDLIfc DPCOperationsTypeLib DPCOperationVBScript GetInput
	GetOutput	GetOutput (CATBSTR pName, CATIABase oValue)	Not supported any longer. 1 replacing method Frameworks Typelibs Objects Methods CATBehaviorIDLIfc DPCOperationsTypeLib DPCOperationVBScript GetOutput
	PutOutput	PutOutput (CATBSTR pName, in CATIABase oValue)	Not supported any longer. 1 replacing method Frameworks Typelibs Objects Methods CATBehaviorIDLIfc DPCOperationsTypeLib DPCOperationVBScript PutOutput
Behavior	TestInput	TestInput (CATBSTR pName, long iExist)	Not supported any longer. 1 replacing method Frameworks Typelibs Objects Methods CATBehaviorIDLIfc DPCOperationsTypeLib DPCOperationVBScript TestInput
	TestOutput	TestOutput (CATBSTR pName, long iExist)	Not supported any longer. 1 replacing method Frameworks Typelibs Objects Methods CATBehaviorIDLIfc DPCOperationsTypeLib DPCOperationVBScript TestOutput
Property Behaviors (Read)	Behaviors	(CATIABehaviors oBehaviors)	Refer to the following article: Business Process Knowledge Template Changes .
BehaviorExtension	Not supported any longer. Refer to the following article: Business Process Knowledge Template Changes . Still available in another framework: CATBehaviorIDLIfc and from the DPCOperationsTypeLib typelib. Some properties or methods are changed:		
Properties or Methods Signatures		Comments	
BehaviorVBScript	Cancel	Cancel()	Refer to the following article: Business Process Knowledge Template Changes .
	Done	Done()	Refer to the following article: Business Process Knowledge Template Changes .
	Start	Start()	Refer to the following article: Business Process Knowledge Template Changes .
	Suspend	Suspend()	Refer to the following article: Business Process Knowledge Template Changes .

CATOBMInterfaces

This table summarizes the V5 CATOBMInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
BehaviorSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
	Use it with the CATBehaviorCATFct setting repository. See the settings section in Automating Settings .

GenKnowledgeInterfaces

This table summarizes the V5 GenKnowledgeInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments	
ExpertRuleBaseRuntime	Still available in another framework: KnowHowIDLIfc and from the GenKwe typelib. Some properties or methods are changed:	
Properties or Methods	Signatures	Comments
ImportFromFile	ImportFromFile (CATBSTR iPath, in boolean iForce)	
	ImportWithLink (CATIABase iRoot, in boolean iForce)	

PCBoardBase

This table summarizes the V5 PCBoardBase Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with 3DEXPERIENCE platform. For each of these:

- If the object still exists as in V5, but in a different framework, its 3DEXPERIENCE platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Moved Objects

This table summarizes the V5 PCBoardBase objects that still exist in **3DEXPERIENCE** platform, but that are located in other frameworks. You can use this table to get their typelib.

Objects	V5 Typelibs	3DEXPERIENCE platform Frameworks	3DEXPERIENCE platform Typelibs
PCBArea	CATPCBoardIDLItf		PCBITF
PCBComponent	CATPCBoardIDLItf		PCBITF
PCBHoleAndPattern	CATPCBoardIDLItf		PCBITF

Changed Objects

Objects	Comments
PCBWorkbench	Not supported any longer. 1 replacing object. CATPCBoardIDLItf PCBIF PCBService
	Frameworks Typelibs Objects

DNBDeviceInterfaces

This table summarizes the V5 DNBDeviceInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
DevAnalysisSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
	Use it with the DNBDeviceBase setting repository. See the settings section in Automating Settings .

DNBRobotInterfaces

This table summarizes the V5 DNBRobotInterfaces Automation objects, properties, and methods that you may have used in your applications built on top of V5 and that do not exist any longer with **3DEXPERIENCE** platform. For each of these:

- If the object still exists as in V5, but in a different framework, its **3DEXPERIENCE** platform typelib is given.
- If a one-to-one mapping exists to replacing objects, properties, or methods, links to these substitutes are listed and pointed to
- Otherwise, an adoption article is pointed to to explain how to replace or get rid of them.

Changed Objects

Objects	Comments
RSSSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
	Use it with the DNBRRSSSettings setting repository. See the settings section in Automating Settings .
RobAnalysisSettingAtt	Not supported any longer. This is a setting interface, replaced with the generic one: SettingRepository .
	Use it with the DNBRobotBase setting repository. See the settings section in Automating Settings .