Agile Product Lifecycle Management

MCAD Connectors for Agile Engineering Collaboration Administration Guide

V3.1.0.0

Oracle Part Number – E38568-01

December 2012

Copyrights and Trademarks

IMPORTANT NOTICE

Contact Address (for Support Services see Preface chapter):

xPLM Solution GmbH
Devrientstr. 5
D - 01067 Dresden, Germany
www.xplm.com

xPLM Solution Inc.
15 Charles Street
Three Rivers, MA 01080 U.S.A.

CONTENTS

Preface

Contacting Oracle Support Services

For Oracle Agile Engineering Collaboration support contact the Oracle Global Customer Support (GCS) via www.oracle.com/support or My Oracle Support via https://support.oracle.com.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

This documentation may contain links to web sites of other companies or organizations that xPLM Solution does not own or control. xPLM Solution neither evaluates nor makes any representations regarding the accessibility of these web sites.

Pro/ENGINEER Connector Administration

This section provides a complete summary of configuration options available for the Pro/ENGINEER connector. Once the basic installation has been done following the instructions in the *Installation Guide*, you can refer here for details of all possible settings.

Note that in addition to the configuration files listed here, the EC Web Connector must be additionally configured to provide complete operation of the Pro/ENGINEER Connector. See the *EC Web Connector Configuration Options* section for details.

Table: List of all Configuration Files for the Pro/E Connector

Configuration files	Purpose	Location
xAcp.cfg	System configuration	<Install Directory>\xacp\com
AcpCustomer9.ini	Mapping and configuration	<Install Directory>\xacp\ini

Note Configuration files typically change content between connector releases. When upgrading to a new release, please incorporate your site's configuration settings into the new version of the configuration files. Failure to do so will cause unpredictable behavior of the connector.

Configuration file xAcp.cfg

The configuration file **xAcp.cfg** contains basic system parameters. It is described fully in the *Installation Guide*, section *Editing the Configuration File* on page 7.

Setup the Attribute Mapping

Use the *Mapping Editor* to define the attribute mapping. Legacy mappings, as described in the following sections, are still supported.

Mapping file AcpCustomer9.ini

This is the main file for controlling the behavior of the Pro/E Connector. This file is structured in several sections. The first line of a section starts with a left square bracket followed by a space and its name again followed by a space and the right square bracket. Each section starts with the section name. A comment line starts with the # sign.

Note Please make sure not to leave blank lines when editing the AcpCustomer9.ini file.

The table below provides descriptions of all of the sections in **AcpCustomer9.ini** file.

Table: Description of all sections in xAcpCustomer9.ini

Section name	Description
Initialize	Common switches to control the behavior of the Pro/E Connector
ProEToAgile.Create	This mapping section is used for initial creation of design objects using the Save command.
ProEToAgile.Update	This section is used when the existing design objects are updated via the Save command.
AgileToProE.ProE	Defines those Agile attributes that are saved automatically into all Pro/E files, during the Save command.
AgileToProE.PRT	Defines those Agile attributes that are saved automatically into Pro/E PRT files, during the Save command.
AgileToProE.DRW	Defines those Agile attributes that are saved automatically into Pro/E DRW files, during the Save command.
AgileToProE.ASM	Defines those Agile attributes that are saved automatically into Pro/E ASM files, during the Save command.
AgileGetProperties.PRT	Defines those Agile attributes that are saved into Pro/E PRT files, when using the Update Properties command.
AgileGetProperties.DRW	Defines those Agile attributes that are saved into Pro/E DRW files, when using the Update Properties command.
AgileGetProperties.ASM	Defines those Agile attributes that are saved into Pro/E ASM files, when using the Update Properties command.

The following table provides details from each section.

Table: [Initialize] Section Parameters

Parameter name in section [Initialize]	Parameter values	Description
AcpDebug	**0** / 1 / 2 / 3	0 → no Debug 1 → write full debuginfo to ...AcpUser\log\acp.log (bad performance) 2 → write additional timer info to Pro/E message log (trail.txt), no acp.log will be generated 3 → write only timer info to ...AcpUser\log\acp.log AcpDebug
JNI_DEBUG	**0** / 1	0 → no JNI Debug 1 → write JNI debug info to ...AcpUser\log\proxy.log
AcpInitialRenameObject	**1** / 0	0 → no Pro/ENGINEER file renaming 1 → Pro/ENGINEER file rename during initial check-in
AcpSaveDrwFrm	**1** / 0	1 → Pro/ENGINEER drawing formats (FRM) are stored in Agile 0 → ignore Pro/ENGINEER drawing formats (FRM)
AcpSaveLay	**1** / 0	1 → Pro/ENGINEER layouts (LAY) are stored in Agile 0 → ignore Pro/ENGINEER layouts (LAY)
AcpHelpPartIdent	**ITEM**	Name of Pro/E parameter used to identify helper parts. These objects are saved into Agile as design objects, but are filtered out when create Item/BOM
AcpHelpPartValue	**N**	Value that the Pro/E parameter should be set to in order to activate the filter

Parameter name in section [Initialize]	Parameter values	Description
AcpHelpPartSkeletonCheck	**1** / 0	1 → Automatically identify skeleton parts as helper parts. Skeleton parts are filtered out when create Item/BOM 0 → use AcpHelpPartIdent / AcpHelpPartValue settings to identify skeleton parts as helper parts
AcpReadFindNumber	**1** / 0	1 → reading "FindNo" during "Update properties" and provide parameter AGILE_FIND_NO for use with Pro/REPORT 0 → not reading "FindNo" during "Update properties"
AcpCreateInterchangeRelation	**1** / 0	1 → create additional relation for components of interchange ASMs during "Save" 0 → not creating additional relation for components of interchange ASMs
AcpLoadUpdateDrwProperties	**1** / 0	1 → automated call of function "Update properties" after load a DRW from Agile 0 → No action after load a DRW from Agile
AcpFamilyCheckVerify = 0/1	**1 / 0**	0 -> Do NOT check family instance verify status before saving 1 -> Check family instance verify status before saving Default entry: 0

Mapping Options for [ProEToAgile.XXXX] Sections

Each mapping consists of a pair of objects. The right side of the pair defines information that can be extracted from Pro/E. Here Pro/E is the source of the attribute value. The left side of the pair defines the attribute value's target location in Agile.

There are several configuration options for the right hand side that define what kind of data should be extracted from Pro/E, and what kind of transformation can be applied to the data. Each right side attribute consists of three sections, for example:

```
DESCRIPTION = Std.ObjectName-Type.ToUpper
```

The first section is either *Std* or *Par*. *Std* refers to Pro/E system attributes such as file name, object type, version of Pro/E that is being used, and so forth.

Table: Standard mapping values using "Std2" prefix

Std.CreSystem	Pro/E version such as "Pro/E Wildfire 5"
Std.VerStamp	Timestamp
Std.FileName	File name, for example "BOLT.PRT"
Std.ObjectName	Pro/E file name without the extension - "BOLT"
Std.ObjectName-Type	Object name with the type appended. This creates an easy way to differentiate an assembly from a part. Examples include: BOLT-PRT, BOLT-ASM, or BOLT-DRW.
Std.ObjectType	Pro/E object type. Possible values are PRT, ASM, DRW, or FRM.

Par is a reference to user-defined parameter in Pro/ENGINEER, such as `MATERIAL`, `DESCRIPTION`, or `ENGINEER`. These types of mappings are only useful where the Pro/ENGINEER file has a parameter corresponding to the name mentioned in the mapping.

Finally, the final suffix is a description of how the data should be modified. The following modifiers are possible:

Table: Suffix Options for Mapping

ToUpper	Transfer all characters to upper case
ToLower	Transfer all characters to lowercase
None	Do not modify the data
Range-<idx1>-<idx2>	Range of the string from position idx1 to idx2, example: Part.PartNumber.Range-0-2
Prefix	Prefix to be added in front of the string, example: Par.PartNumber.PrefixPRT

Suffix	Suffix to append to the string, example: Par.PartNumber.SuffixPRT

There are two special values that are used on the left side of these mappings. In the [ProEToAgile.Create] section, you use the value `CAX_NEW_NUMBER` to represent the number field that will be assigned to the newly created Design object.

The following example maps a Pro/ENGINEER parameter `NAME` to the Agile attribute `DESCRIPTION` and the Pro/ENGINEER version to Agile attribute `CAX_CRE_SYSTEM`:

Table: Example Mapping Definitions

DESCRIPTION	Par.NAME.None
CAX_CRE_SYSTEM	Std.CreSystem.None

Mapping Options for [AgileToProE.XXXX] Sections

These sections are used to define mappings from Agile to Pro/E which occur automatically during the save process. As this will add time to the save process, the list of attributes should be kept to the bare minimum that absolutely needs to be kept synchronized. Other attributes can be synchronized using *Update Properties* as described in the next section.

The format of this section is:

```
DocNumber    =    NUMBER
```

Where the left side value is the name of the Pro/E parameter to be updated, and the right side is the Agile attribute value to be used as the source.

Mapping Options for [AgileGetProperties.XXX] Sections

These sections are used to define mappings from Agile to Pro/E, which occur when the user runs the *Update Properties* command manually. For standard attributes the format of this section is:

```
CAD Parameter  =     <Source Table Field>.Format
```

For example:

```
Agile_Des      =     DESCRIPTION.ToUpper
```

Where the left side value is the name of the Pro/E parameter to be updated, and the right side is the Agile attribute value to be used as the source.

For part history and change history attributes, which are arranged in a table, the format of this section is:

```
CAD Parameter = <Filter Table>_Field,<Filter Value>,<Filter>,<Source
Table>_Field.Format
```

For example:

```
Agile_CreUser = History_Action,Create,first,History_User.None

HIS_RELDATE_1 = Change History_Status,Released,last,Change History_Rel
Date_int.Date01
```

Where the left side value is the name of the Pro/E parameter to be updated, and the right side specifies how to find the desired row and column in the table below:

Section	Represents	Example
<Filter Table>	Agile tab name to search	Title Block
Field	Desired column to search	Action
<Filter Value>	Value to detect in the column	Create
<Filter>	Which row to select, with these options: first first+n n=integer value last last-n n=integer value	first
<Source Table>	Agile tab name to retrieve value from	History
Field	Desired column to retrieve value from	User

Section	Represents	Example
Format	Text processing	None

Options for "Format"

The Format string allows you to perform additional processing on the text string being passed back into CAD. This includes predefined formats and general TCL format procedures.

Predefined formats

Format	Description
None	no processing
ToLower	convert the value to lower case
ToUpper	convert the value to upper case
Range-x-y	substring of the value from index x to index y (y may be numeric or "end")
Date01	convert int dateformat to "%d.%m.%y %H:%M:%S" example: 01.01.2007 00:00:00
Date02	convert int dateformat to "%d.%m.%Y" example: 01.01.2007
Date03	convert int dateformat to "%d.%m.%y" example: 01.01.07
Date04	convert int dateformat to "%d-%m-%y" example: 01-01-07
Date05	convert int dateformat to "%m/%d/%y" example: 01/01/07
Date06	convert int dateformat to "%d-%b-%y" example: 01-Jan-07
Prefix<str>	append a prefix <str> to the value
Suffix<str>	append a suffix <str> to the value

TCL format procedures

Any registered (tclIndex) TCL procedure that gets the current value as input and returns the formatted string. For instance:

```
proc MyFormat { value } {
  set formatedvalue $value
  return $formatedvalue
}
```

Mapping Part Attributes

In addition to mapping attributes from the CAD document back into CAD, you can map attributes from the corresponding Part object that has been associated to the Design object. In order to specify a part attribute, simply prefix the attribute value with `PART:`. This example shows how to map both the Document Number and Part Number into CAD:

```
Agile_DocId     =  NUMBER.None
Agile_PartId    =  PART:NUMBER.None
```

CATIA V5 Connector Administration

This section provides a complete summary of configuration options available for the CATIA V5 connector. Once the basic installation has been done following the instructions in the *Installation Guide*, you can refer here for details of all possible settings.

Note that in addition to the configuration files listed here, the EC Web Connector must be additionally configured to provide complete operation of the CATIA V5 Connector. See the *EC Web Connector Configuration Options* section for details.

Table: List of all Configuration Files for the CATIA V5 Connector

Configuration file	Purpose	Location
Acc.cfg	System configuration	<Install Directory>\acc\com
AccInitialize.ini	Configuration	<Install Directory>\xacc\ini
AccCustomer9.ini	Mapping	<Install Directory>\xacc\ini

Note Configuration files typically change content between connector releases. When upgrading to a new release, please incorporate your site's configuration settings into the new version of the configuration files. Failure to do so will cause unpredictable behavior of the connector.

Configuration file Acc.cfg

The configuration file **Acc.cfg** contains basic system parameters. It is described fully in the *Installation Guide*, Section *Editing the Configuration File* on page 14.

Configuration file AccInitialize.ini

This is the main file for controlling the behavior of the CATIA V5 Connector. This file has a single [Initialize] section. A comment line starts with the # sign.

Note Please make sure not to leave blank lines when editing the file.

Table: [Initialize] Section Parameters

Parameter name in Section [Initialize]		Parameter values	Description
AccCustomerId	=	None	System setting (do not change)
AccLanguage	=	**English**	Language setting

Parameter name in Section [Initialize]		Parameter values	Description
AccMappingFile	=	Acc.ini	Mapping file name
AccCustomerFile	=	AccCustomer9.ini	Customer file name
AccMessages	=	AccMessages.ini	Messages file name
AccDebug	=	**1** / 0	Turns debug mode on (1) and off (0). A log file is written to the user's working directory.
AccHelpPartIdent	=	ITEM	Name of CATIA V5 property used to identify models in the design that should not be included in the BOM. These objects are saved into Agile as Documents, but are filtered out when using the Create Item/BOM function.
AccHelpPartValue	=	NO	Value that the CATIA V5 property should be set to in order to activate the filter.
AccAgileBackupId	=	AgileID	Indicates the field to use for re-associating a file to the correct Agile Document. This assignment tracks the Agile Document number.
AccAgileBackupName	=	AgileName	Indicates the field to use for re-associating a file to the correct Agile Document. This assignment tracks the Agile filename.
AccEnableRename	=	1	0 = files are not renamed 1 = files are renamed to match the Agile Number field or custom mapping
AccSchemeOfFileName	=	%	Format definition (in "C" style) used to define the CATIA filename
AccFileNameValues	=	NUMBER / CATIAFILE	Basis of the filename. Standard values are either NUMBER (Agile Document Number) or CATIAFILE (original filename)

Filename creation

During the first save into Agile, a new CATIA V5 filename can be created. In the file **AccInitialize.ini** are two variables that control this process:

- AccFilenameValues
- AccSchemeOfFileName

AccFilenameValues can contain a list of attributes from Agile either defined in the EC Web Connector definition file or simply `CATIAFILE`. `CATIAFILE` means the usage of the original Catia file name. AccSchemeOfFileName is a format definition based on the *C style*.

```
#
AccSchemeOfFileName  = %s
AccFileNameValues  = NUMBER
```

After check in of a part to Agile, the object will be renamed to **D00444.CATPart** because D00444 is the number of the Agile document.

```
#
AccSchemeOfFileName  = %s
AccFileNameValues  = CATIAFILE
```

After check in of a part to Agile the object will not be renamed.

```
#
AccSchemeOfFileName  = CAT-%s
AccFileNameValues  = NUMBER
```

After check in of a part to Agile the object will be renamed to **CAT-D00444.CATPart**.

[Customer Functions] Section

To better support the ability for project-based customization of TCL scripting, entry points are now provided for TCL add-ins through the [CustomerFunctions] section in **AccInitialize.ini**.

[CustomerFunctions]

...

```
<EntryPoint> = <Customer specific procedure>
```

....

There are 7 predefined entry points:

1. CatiaScanTree-01
2. CatiaScanTree-02
3. CatiaScanTree-03

4. CatiaAccSaveToAgile-01
5. CatiaAccLoad-01
6. CatiaAccSave-01
7. CatiaAccUpdateFrame-01

Setup the Attribute Mapping

Please use the Mapping Editor to define the attribute mapping. The legacy mappings as described in the following sections is still supported.

Mapping file AccCustomer9.ini

This is the main file for controlling attribute mapping in the CATIA V5 Connector. This file is structured in several sections. The first line of a section starts with a left square bracket followed by a space and its name again followed by a space and the right square bracket. Each section starts with the section name. A comment line starts with the # sign.

Note Please make sure not to leave blank lines when editing the file.

The following table gives a description of all sections in **AccCustomer9.ini**, and the following tables provide the details of each section.

Table: Description of all sections in AccCustomer9.ini

Section name	Description
CatiaToAgile.DOCUMENT	This mapping section is used for assigning attributes when Documents using the *Save* command.
CatiaToAgileUpdate.DOCUMENT	This mapping section is used for assigning attributes when updating Documents using the *Save* command.
CatiaToAgile.FILEFOLDER	OBSOLETE
CatiaToAgile.ITEM	This mapping section is used for creating and updating Parts using the *Create Item/BOM* command.
AgileTo.Catia	Defines those Agile attributes that are saved automatically into all CATIA V5 files, during the *Save* command.

Section name	Description
AgileTo.CATPart	Defines those Agile attributes that are saved automatically into CATIA V5 CATPart files, during the *Save* command.
AgileTo.CATDrawing	Defines those Agile attributes that are saved automatically into CATIA V5 CATDrawing files, during the *Save* command.
AgileTo.CATProduct	Defines those Agile attributes that are saved automatically into CATIA V5 CATProduct files, during the *Save* command.
AgileGetProperties.Catia	Defines those Agile attributes that are saved into all CATIA V5 files, when using the *Update Properties* command.
AgileGetProperties.CATPart	Defines those Agile attributes that are saved into CATIA V5 CATPart files, when using the *Update Properties* command.
AgileGetProperties.CATDrawing	Defines those Agile attributes that are saved into CATIA V5 CATDrawing files, when using the *Update Properties* command.
AgileGetProperties.CATProduct	Defines those Agile attributes that are saved into CATIA V5 CATProduct files, when using the *Update Properties* command.
FrameDefinition	Defines those Agile attributes that are mapped onto drawing title blocks, when using the *Update Title Block* command.
AccCreateObjectTypes	Not used
CatiaToAgileNew.DOCUMENT	This mapping section is used for assigning attributes when creating Documents using the *New* command.
AccSaveViewable.CATPart	Defines types of viewable files that can be saved for CATParts in the *Save With…* command
AccSaveViewable.CATProduct	Defines types of viewable files that can be saved for CATProducts in the *Save With…* command
AccSaveViewable.CATDrawing	Defines types of viewable files that can be saved for CATDrawings in the *Save With…* command

Mapping Options for [CatiaToAgile.XXXX] Sections

Each mapping consists of a pair of objects. The right side of the pair defines information that can be extracted from CATIA V5. Here, CATIA V5 is the source of the attribute value. The left side of the pair defines the attribute value's target location in Agile.

There are several configuration options for the "right side" that define what kind of data should be

extracted from CATIA V5 and what kind of transformation can be applied to the data. Each right side attribute consists of three sections, for example:

```
DESCRIPTION    =  Std.DescriptionReference.ToUpper
```

The first section is either *Std*, *Par*, or *Def*. *Std* refers to CATIA V5 system attributes, as listed here:

Table: Standard mapping values using "Std" prefix

Std.DescriptionReference
Std.Extension
Std.PartNumber
Std.Definition
Std.Nomenclature
Std.Revision

Par is a reference to user-defined property in CATIA V5, such as `MATERIAL`, `DESCRIPTION`, or `ENGINEER`. These types of mappings are only useful where the CATIA V5 file has a property corresponding to the name mentioned in the mapping.

Def is a default fixed string value.

Finally, the final suffix is a description of how the data should be modified. The following modifiers are possible:

Table: Suffix Options for Mapping

ToUpper	Transfer all characters to uppercase
ToLower	Transfer all characters to lowercase
None	Do not modify the data
Range-<idx1>-<idx2>	Range of the string from position idx1 to idx2, for example: Part.PartNumber.Range-0-2
Prefix	Prefix to be added in front of the string, for example: Par.PartNumber.PrefixPRT
Suffix	Suffix to append to the string, for example: Par.PartNumber.SuffixPRT

There are two special values that are used on the left side of these mappings. In the [CatiaToAgile.DOCUMENT] section, you use the value `CAX_NEW_NUMBER` to represent the Number field that will be assigned to newly created Documents. In the [CatiaToAgile.ITEM] section, you use the value `ITEM` to represent the Number field that will be assigned to newly created Parts.

Mapping Options for [AgileTo.XXXX] Sections

This section is used to define mappings from Agile to CATIA, which occur automatically during the save process. As this will add time to the save process, the list of attributes should be kept to the bare minimum that absolutely needs to be kept synchronized. Other attributes can be synchronized using *Update Properties*, as described in the next section. For formatting details, see *Mapping Options for Update Properties Sections – CATIA*.

Mapping Options for [AgileGetProperties.XXX] Sections

This section is used to define mappings from Agile to CATIA V5, which occur when the user runs the *Update Properties* command manually. For formatting details, see *Mapping Options for Update Properties Sections – CATIA*.

Mapping Options for [FrameDefinition] Section

This section is used to define mappings from Agile attributes to the CATIA V5 drawing title block, which occurs when the user runs the *Update Title Block* command. For formatting details, see *Mapping Options for Update Properties Sections – CATIA*.

Mapping Options for Update Properties Sections – CATIA

Multiple sections of the **AccCustomer9.ini** file, as listed above, are used to define mappings from Agile to CATIA. For standard attributes the format of this section is:

```
CAD Parameter = <Source Table>_Field.Format
```

For example:

```
Agile_Des = Title Block_Description.ToUpper
```

Where the left side value is the name of the CATIA parameter to be updated. For the [AgileTo.XXXX] and [AgileGetProperties.XXX] sections, the formatting of the left side matches the description shown for the RIGHT side of the [CatiaToAgile.XXXX] section (see above for details). For the [FrameDefinition] section, the left side represents a CATIA text property in the format Text.n, where n is an integer.

The right side can be either the symbolic attribute name from the **CaxClient.xml** file (such as `NUMBER`, `DESCRIPTION`, etc.) or any Agile attribute represented as follows:

Section	Represents	Example
<Source Table>	Agile tab name	Title Block
Field	Agile attribute name	Description

Section	Represents	Example
Format	Text processing	ToUpper

For history and change history attributes, which are arranged in a table, the format of this section is:

```
CAD Parameter = <Filter Table>_Field,<Filter Value>,<Filter>,<Source Table>_Field.Format
```

For example:

```
Agile_CreUser = History_Action,Create,first,History_User.None
HIS_RELDATE_1 = Change History_Status,Released,last,Change History_Rel Date_int.Date01
```

Where the left side value is the name of the CATIA parameter to be updated and the right side specifies how to find the desired row and column in the table below:

Section	Represents	Example
<Filter Table>	Agile tab name to search	Title Block
Field	Desired column to search	Action
<Filter Value>	Value to detect in the column	Create
<Filter>	Which row to select, with these options: first first+n n=integer value last last-n n=integer value	first
<Source Table>	Agile tab name to retrieve value from	History
Field	Desired column to retrieve value from	User
Format	Text processing	None

Options for "Format"

The Format string allows you to perform additional processing on the text string being passed back into CAD. This includes predefined formats and general TCL format procedures.

Predefined formats

Format	Description
None	no processing
ToLower	convert the value to lower case
ToUpper	convert the value to upper case
Range-x-y	substring of the value from index x to index y (y may be numeric or "end")
Date01	convert int dateformat to "%d.%m.%y %H:%M:%S" example: 01.01.2007 00:00:00
Date02	convert int dateformat to "%d.%m.%Y" example: 01.01.2007
Date03	convert int dateformat to "%d.%m.%y" example: 01.01.07
Date04	convert int dateformat to "%d-%m-%y" example: 01-01-07
Date05	convert int dateformat to "%m/%d/%y" example: 01/01/07
Date06	convert int dateformat to "%d-%b-%y" example: 01-Jan-07
Prefix<str>	append a prefix <str> to the value
Suffix<str>	append a suffix <str> to the value

TCL format procedures

Any registered (tclIndex) TCL procedure that gets the current value as input and returns the formatted string. For instance:

```
proc MyFormat { value } {
  set formatedvalue $value
  return $formatedvalue
}
```

Mapping Part Attributes

In addition to mapping attributes from the CAD document back into CAD, you can map attributes from the corresponding Part object that has been associated to the document using the *Create Item/BOM* command. In order to specify a Part attribute, simply prefix the attribute value with `PART:`. This example shows mapping both the Document Number and Part Number into CAD:

```
Agile_DocId   =  Title Block_Number.None
Agile_PartId  =  PART:Title Block_Number.None
```

SolidWorks Connector Administration

This section provides a complete summary of configuration options available for the SolidWorks connector. Once you have completed the basic installation based on instructions in the *Installation Guide*, you can refer here for details of all possible settings.

Note that in addition to the configuration files listed here, the EC Web Connector must also be configured to provide complete operation of the SolidWorks Connector. See the *EC Web Connector Configuration Options* section for details.

Connector Configuration Settings

The configuration of the connector is done in XML files, which are located in the **components\xml** subdirectory of the integration. The following files are important for the connector:

XPlmSolidWorksConnector.xml – base configuration of the SolidWorks connector

XPlmSWAgileAddin.xml – contains the Agile menu definition and Addin registration. **This file should not be changed**.

xPLMAgile9SolidWorksTransaction.xml – contains the configured transactions for the Agile load and save processes. **This file should not be changed**.

Renaming, Configuration Handling and Options

This section describes the available settings and valid values for the connector in **XPlmSolidWorksConnector.xml**.

Attention: Do not enable the RenameOnSave or RenameOnInitialSave options if you use big assemblies, suppressed components, or external references. Rely on the RenameOnLoad feature in

these cases. Otherwise, references and assemblies may be destroyed during save if not all related components are loaded or available.

Setting	Purpose and available values
RenameOnLoad	NUMBER – rename file names equally to the PLM number CAX_FIL_NAME – don't rename on load Any other value **Default: CAX_FIL_NAME**
RenameOnInitialSave	true – rename files on initial save false – no renaming on save **Default**: **false** **ATTENTION: Do not use this option if you use large assemblies or external references and suppressed components. Use the RenameOnLoad feature in these cases!**
RenameOnSave	true – rename files on save as false – no renaming on save **Default**: **false** **ATTENTION: Do not use this option if you use large assemblies or external references and suppressed components. Use the RenameOnLoad feature in these cases!**
RenamingRule	Renaming rule for building the filename **Default %CAX_NEW_NAME%**
ConfiguredDefault	Controls the default behaviour for handling configurations. If set to true, each configuration is treated as a separate Design object. If set to false, no Design object is created for each configuration. The default can be overridden by the file

	property set in the ConfiguredProperty. **Default: true**
ConfiguredProperty	If the given property name is contained in a file and set to No then no configurations are created in PLM. **Default**: **Configured**
ConfiguredProperty2	Additional property name to identify configured files. If property name is contained in a file and set to No then no configurations are created in PLM. **Default**: **Configured**
ConfiguredValue_Configured	Optional value of the ConfiguredProperty or ConfiguredProperty2 that would be interpreted as “Yes” (configured).
ConfiguredValue_NotConfigured	Optional value of the ConfiguredProperty or ConfiguredProperty2 that would be interpreted as “No” (not configured).
MasterConfigProperty	If the given property name is contained in the configuration specific properties and the value of this property points to an existing configuration in the same file, the linked configuration is used and no extra configuration object is created in PLM. **Default**: **MasterConfig**
EnableSolidWorksLogging	false - logging disabled true - logging enabled **Default**: **false**
SolidWorksLogFile	Value is the full path to a logfile, required if logging is enabled e.g. C:\caxlog\SolidWorks.log
EnableScriptEngineLogging	false - logging disabled true - logging enabled

	Default: **false**
ScriptengineLogFile	Value is the full path to a logfile, required if logging is enabled. If you do not specify a path the log file is written to the user home AgileCache folder. e.g. C:\caxlog\xacw.log **Default: xacw.log**
SolidWorksMenuFiles	The AddIn Menu file in the xml directory **Default**: **XPlmSWAgileAddin.xml**
SolidWorksStandardPartDir	Directory root under which parts are detected as standard parts **Default: C:\SolidWorks Data**
SolidWorksCreateUniqueFileNames	true – create unique file names false – no special logic **Default**: **true**
SolidWorksUseLocalFileCache	true – use local cache false – no local cache **Default**: **true**
AllowRecursiveStructure	true – allow transfer of recursive structure false – no recursing structure **Default**: **true**
SWAddins	Additional addins to be loaded **Default**: **empty**
IgnoreMissingParts	true – ignore missing parts false – throw error for missing parts **Default**: **true**
SolidWorksCreateBitmapPreview	true – create preview bitmap

	false – no special logic **Default**: **false**
SolidWorksCommandTabName	String to display **Default**: **xPLM Solution**
SolidWorksScriptEngine	Do not change **Default**: **intern**
SolidWorksEvent_StartNotify	true – startup integration immediately to preserve memory false – start integration on demand **Default**: **true**
FindPLMObject	Whether to search for existing configurations and files in PLM. Allowes values are true and false. **Default: false**
SolidWorks_DisableUpdateDrawingBOM	Whether to reset the update flag in drawings to suppress updating the Partslist on Load. Allowes values are true and false. **Default: false**
AppendPLMFieldsToViewableNames	Whether to append additional PLM fields into the viewables filenames. Allowes values are true and false. **Default: false**
AppendingRuleViewables	FieldID or CAX fieldnames to append. Format like %REVISION%
SolidWorks_BulkLoad_AllConfigurations	Bulkloaded switch to create all or only use configurations on initial import. Allowes values are true and false. **Default: false**
SolidWorks_InstanceViewables	Viewable type to create for Configurations that have no real 3D model file. **Attention: if you set this option, depending on the assembly complexity the generation is very time and**

	resource consuming. It is not recommended to use this in big assemblies because Solidworks needs to regenerate each saved configuration, which can lead to unstable CAD behaviour. Allowes values are X_T and false. **Default: false**
SolidWorks_AlwaysExtractExternalReferences	true – external references are always traversed false – external references are traversed on demand with user prompt only (Default) **Default: false**

Setup the Workspace Root

The Java environment and workspace root are set in **xacw\components\ini\start_acx.bat**

Configure the workspace root by setting these values:

```
set cax_temp=C:\AgileEC\wspaces\Default
set CAX_WORKSPACE_ROOT=C:\AgileEC\wspaces
```

Setup the Java Environment and Workspace Root

The Java Environment is set in **xacw\components\ini\start_acx.bat**

Usually there is no need to modify the Java settings that are delivered with the connector. The script detects the system architecture and initializes the right JRE in **xacw_ini.bat** during the initial registration and also on startup of the connector.

Setup the Attribute Mapping

Please use the *Mapping Editor* to define the attribute mapping.

Solid Edge Connector Administration

This section provides a complete summary of configuration options available for the Solid Edge connector. Once the basic installation has been done following the instructions in the *Installation Guide*, you can refer here for details of all possible settings.

Note that in addition to the configuration files listed here, the EC Web Connector must be additionally configured to provide complete operation of the SolidWorks Connector. See the *EC Web Connector Configuration Options* section for details.

Connector Configuration Settings

The configuration of the connector is done in XML files, which are located in the **components\xml** subdirectory of the integration. The following files are important for the connector:

XPlmSolidEdgeConnector.xml – base configuration of the Solid Edge connector

XPlmSEA9Addin.xml – contains the Agile menu definition and Addin registration. **This file should not be changed.**

XPlmAgile9SolidEdgeTransaction.xml – contains the configured transactions for the Agile load and save processes. **This file should not be changed.**

Renaming, Configuration Handling and options

This section describes the available settings and valid values for the connector in **XPlmSolidEdgeConnector.xml**.

Setting	Purpose and available values

SolidEdge_RenameOnLoad	NUMBER – rename file names equally to the PLM number CAX_FIL_NAME – don't rename on load, use value in CAD filename field **Default: CAX_FIL_NAME**
EnableSolidEdgesLogging	False: Logging disabled True: Logging enabled **Default**: **false**
SolidEdgeLogFile	Value is the full path to a logfile, required if logging is enabled e.g. C:\caxlog\SolidWorks.log
SolidEdge_EnableScriptEngineLogging	False: Logging disabled True: Logging enabled **Default**: **false**
SolidEdge_ScriptengineLogFile	Value is the full path to a logfile, required if logging is enabled. If you don't specify a path the log file is written to the user home AgileCache folder. e.g. C:\caxlog\xace.log **Default: xace.log**
SolidEdgeMenuFiles	The AddIn Menu file in the xml directory **Default**: **XPlmSWAgileAddin.xml**
SolidEdgeStandardPartDir	Directory root under which parts are detected as standard parts **Default**: C:\SolidEdge Data
SolidEdgeIgnoreMissingParts	true – ignores missing files false – error on missing files **Default**: **true**

SolidEdgeEvent_StartNotify	true – startup integration immediately to preserve memory false – start integration on demand **Default**: **true**
SolidEdge_SuppressUnusedMembers	true – suppress unused members in normal save false – show all available members on normal save **Default**: **true**
SolidEdge_SuppressTemplateMemberLinks	true – suppress circular reference from template to all members false – show external reference from template to all members. Needs SuppressUnusedMembers to be set to false as well **Default**: **true**
SolidEdge_RenameOnInitialSave	NOT SUPPORTED
SolidEdge_RenameOnSave	NOT SUPPORTED

Setup the Workspace Root

The Java environment and workspace root are set in **xace\components\ini\start_ace.bat**

Configure the workspace root by setting these values:

```
set cax_temp=C:\AgileEC\wspaces\Default
set CAX_WORKSPACE_ROOT=C:\AgileEC\wspaces
```

Setup the Java Environment and Workspace Root

The Java environment is set in **xace\components\ini\start_ace.bat**

Usually there is no need to modify the Java settings that are delivered with the connector. The script detects the system architecture and initializes the right JRE in start_ace.bat during the initial registration and also on startup of the connector.

Setup the Attribute Mapping

Please use the *Mapping Editor* to define the Attribute Mapping.

Solid Edge has different property pages which are treated transparently by the integration. The properties are read and written to the Custom page with the following exceptions.

CAD Property Name	Solid Edge Property Page
Title	Summary
Subject	Summary
Author	Summary
Keywords	Summary
Comments	Summary
Last Author	Summary
Username	Extended Summary
Document Number	Project
Revision	Project
Project Name	Project
Category	Document
Company	Document
Manager	Document

ATTENTION: The name of the CAD properties below is language specific to the language of your CAD system. In case you use a non English Solidedge you can enable the Scriptenginelogging in order to check the available CAD property names in $HOME\AgileCache\xace.log. Search for this section in the logfile, the available names are seperated by semicolons:

```
SolidedgeProperties - Properties are language specific to the CAD system
language

SolidedgeProperties - Non Custom Properties must be mapped to one of the
following indentifiers

SolidedgeProperties - ProjectInformation - ;Document Number;Revision;Project
Name;
```

```
SolidedgeProperties - DocumentSummaryInformation - ;Category;Presentation
Format;…

SolidedgeProperties - SummaryInformation - ;Title;Subject;Author;…

SolidedgeProperties - ExtendedSummaryInformation - ;Name of Saving
Application;DocumentID;…
```

EC Web Connector Administration

This section provides a complete summary of configuration options available for the EC Web connector. Once the basic installation has been done following the instructions in the *Installation Guide*, you can refer here for details of all possible settings.

Preferences Settings on MCAD-CONFIG folder

The preferences are stored in PLM in a Design filefolder called MCAD-CONFIG. The user needs the Administrator role in PLM assigned in order to update the template **Attributes.xml**.

The **Attributes.xml** is stored locally first and only if you are an administrator the template will be updated and uploaded to PLM. This is done using the save button in the preferences panel. You can also reset the template manually by checking out the MCAD-CONFIG filefolder in web client, adding your local **Attributes.xml** to the files tab, and checking the filefolder back in after upload. The next time a user logs in, the new template will be downloaded. In case the system can't generate the MCAD-CONFIG filefolder automatically, create a design object with this name in the PLM system.

To lock an entry from user modification, you need to edit the **Attributes.xml**. The template in PLM must also be replaced manually. You need to search for a section "GeneralDefaults", there are several FieldCollections contained. Each collection describes one default. There are 3 fields with name/value pairs for each default:

Name	Value

CAX_NAME	Internal setting name (for instance, DesignClass)
Default	The default setting as a string
Editable	true/false (whether the preference is editable or not). If you set to editable=false, even the administrator will get a setting, which is not editable anymore.

Note that the preferences are saved locally first in any case (it is saving to the latest version of **Attributes.xml** in the users home AgileCache folder, for instance **Attributes.xml.8**). If the upload fails due to missing privileges, you can still upload to the MCAD-CONFIG manually and set the defaults properly. The file on the MCAD-CONFIG filefolder must be named **Attributes.xml**. Don't forget to remove the version extension of the local file in this case.

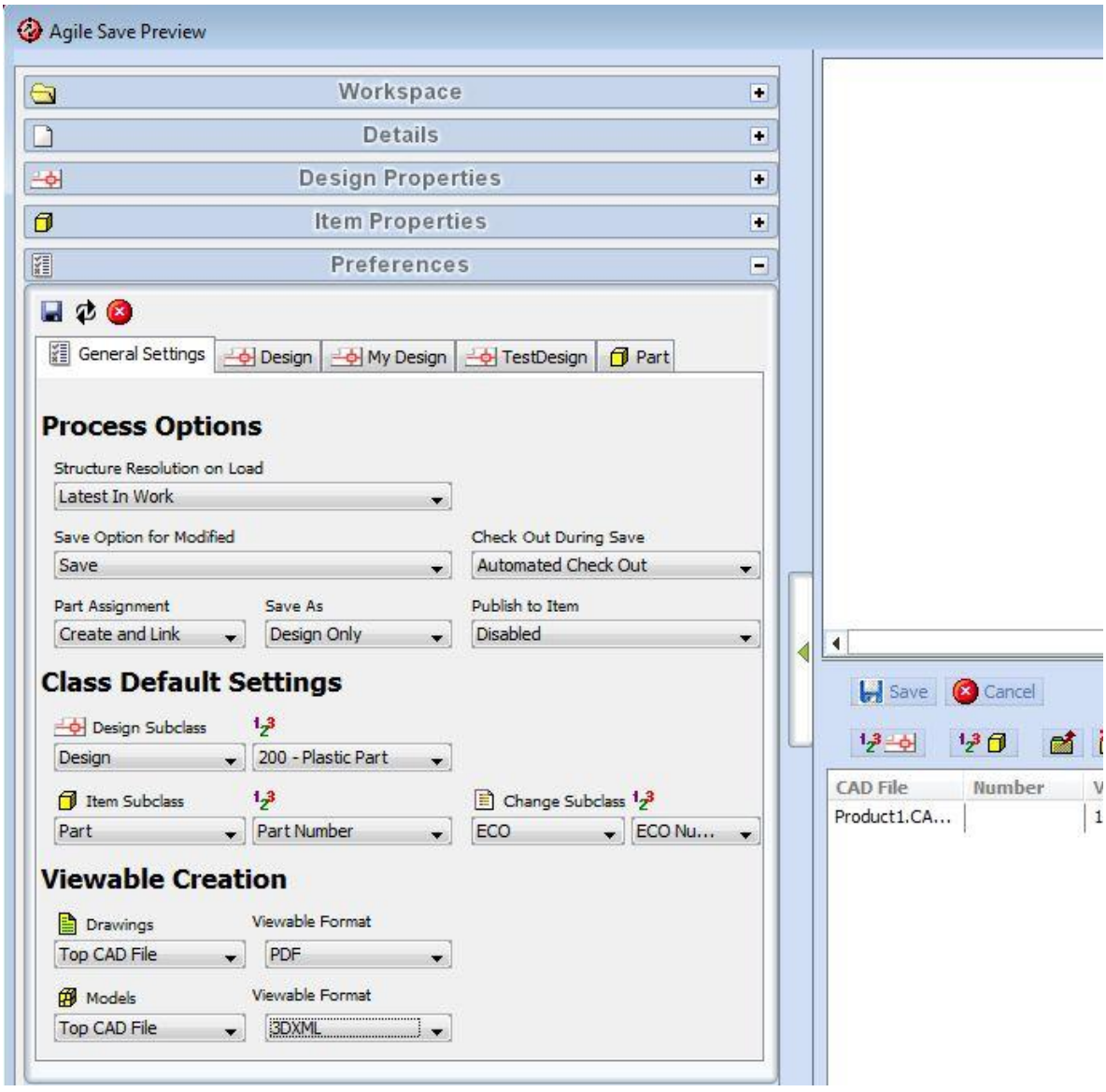
The following table lists the valid preference settings in **Attributes.xml**:

CAX_NAME	Purpose
DesignClass	Default Design Subclass
PartClass	Default Part Subclass
ChangeClass	Default Change Subclass
DesignAutonumber	Default Design Autonumber Source
PartAutonumber	Default Part Autonumber Source
ChangeAutonumber	Default Change Autonumber Source
LoadOption	Default Structure resolution on Load 0 – Latest in Work 1 – Latest Checked In 2 – Latest Published 3 – As Saved
SaveOption	Default Save Option 0 – Save 1 – Checkin 2 – Increment
SaveCheckoutOption	Default Save behavior 0 – Force User Checkout 1 – Automated Checkout
SaveAsOption	Save As behavior for Autonumbers and Design/Part 0 – Design only 1 – Part and Design

ItemOption	Item Creation and Linking option 0 – Create and Link 1 – Update and Link 2 – Link only 3 – Disabled
PublishOption	BOM and Attachments Publishing 0 – Disabled 1 – BOM and Attachments 2 – BOM 3 – Attachments
DrawingViewableCreation ModelViewableCreation	Viewable Creation Option 0 – All CAD Files 1 – Top CAD File 2 – disabled
DrawingViewables ModelViewables	Selected viewables to create during save

Preferences Dialog

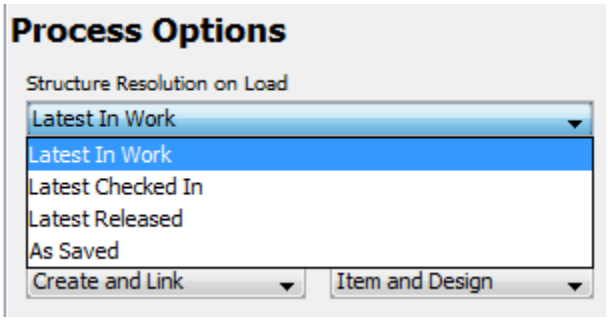
The Preferences dialog is accessed using the button in the Save or Load dialogs or by expanding the left sidebar and the contained preferences container.



The Process Options define the behavior during load and save operations. The Class Default Settings predefine the default subclasses and autonumbers to be used if new Parts, Designs or Change Orders are created.

Load Preferences

The default structure resolution on load is configured using the *Structure Resolution on Load* preference. This defines which versions of children in design structures will be used in an assembly.

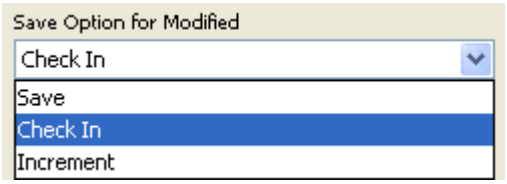


The valid values and their meanings are:

Latest in Work	Select the latest possible design version of a component, including versions that are currently checked out by the current user.
Latest Checked In	Select the latest checked-in design version of a component.
Latest Released	Select the latest design version, which is attached to a released part.
As Saved	Select the design version that was saved within the parent assembly.

Save Preferences

Default Save Option

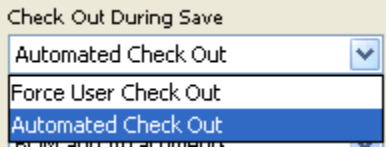


The default save option for modified files defines the preselected option in the save dialog. The valid options are:

Save	The file will be saved into the currently checked out design version. The design remains checked-out after save.
Check In	The file will be saved into the currently checked out design version and then the design will be checked-in.

Increment	The file will be saved into the currently checked out design version and then the design will be checked-in and then checked out again immediately. This way the design remains checked out after save with an incremented version.

Checkout during Save



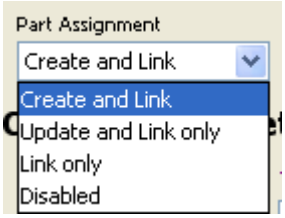
The checkout behaviour during save is controlled by this switch. The valid options are:

Automated Check Out	The design will be checked-out automatically, when it is saved to PLM.
Force User Check Out	The user has to check-out the design in order to be able to save.

Item and Publish Preferences

In order to achieve a transparent Part/Design creation and linking process, some defaults are required to control the simultaneous creation of Parts and Designs.

Part Assignment



The *Part Assignment* controls whether or not Parts will be created simultaneously. The options and their meanings are:

Create and Link	This will create new Part objects, if a new Design is created. The Part will be linked to the Design and the Part properties will be updated.
Update and Link only	This will not create Parts, but existing Parts will be linked to the Design and the Part properties will be updated.
Link only	This will not create or update Parts. Only the relationship link between the part and the design will be created.

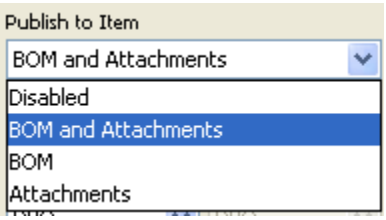
Disabled	Part assignment or creation is disabled completely.

Save As Behaviour

The *Save As Option* controls, whether or not Parts will be created during initial save and save as of a CAD model. The valid options and their meaning are:

Design Only	No Part objects will be created.
Part and Design	Autonumber will be used as the basis for new Design objects, with the CAD extension appended. Part objects are created if the *Part Assignment option* is set to *Create and Link*.

Publish Behaviour



The *Publish to Item Option* controls whether or not a Part BOM is created or updated, and the Design files are attached to the Part objects, after the Design is checked in. The valid options and their meanings are:

Disabled	No Part BOM will be updated and no attachments will be updated.
BOM and Attachments	Part BOM will be updated and the Design files will be attached to the Parts.
BOM	Part BOM will be updated. No Design files will be attached.
Attachments	No Part BOM will be updated. Design files will be attached.

Class Preferences

This section defines the default subclasses and default autonumber sources for all Parts, Designs and Change orders created by the CAD integration. These settings are mainly used in save use cases.

Viewable Creation Preferences

Viewable File Creation Preferences determine the types of viewable files that will be automatically created and attached in PLM along with the native file. This can be set independently for Drawings and Models (parts and assemblies), and can be set to generate the viewable files for all CAD files, only the top CAD file, or no CAD files. Also note that depending on the CAD system, additional configuration work may be necessary to automatically create the viewable files. The available viewable types are defined in the **<Install Directory>\ini\CAXConfig.xml** in the viewables structure:

```
…
<Structure>
   <Name>Viewables</Name>
      <FieldCollection>
         <Field><Name>ViewablesDrawing</Name><Value>PDF;TIF;CGM</Value></Field>
         <Field><Name>ViewablesModel</Name><Value>CGR;WRL;STEP;IGES;3DXML;JT</Value></Field>
      </FieldCollection>
</Structure>
…
```

Basicly, there are two kinds of viewables: Viewables of drawings and viewables of models. Viewable creation for both types is controlled in the *Viewable Creation* section of the Preferences Panel in the Sidebar.

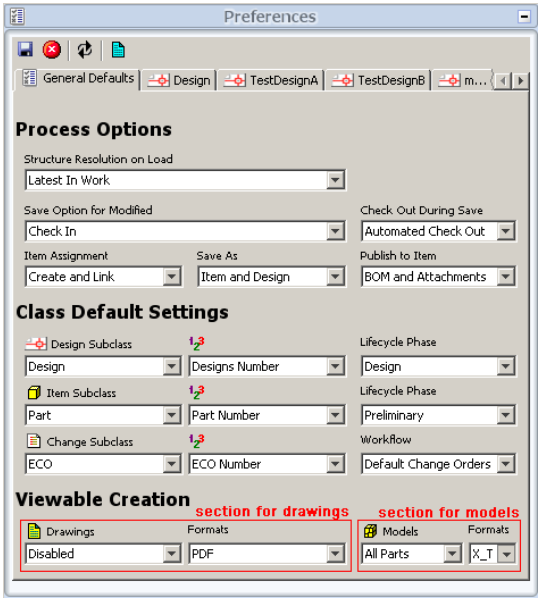


Figure: Preferences Panel. Viewable Creation section is situated on the bottom.

The two combo boxes on the left hand side are used for drawing viewables, the combo boxes on the right hand side for models. The creation of viewable files can be enabled by clicking on one of the *Formats* combo boxes, in the editor window that opens viewable formats can be chosen among the supported formats of the user's CAD.

The combo boxes titled *Drawings* and *Models* define in wich situations the viewables selected in the *Formats* combo boxes should be created.

The following options are available in the *Drawings* combo box:

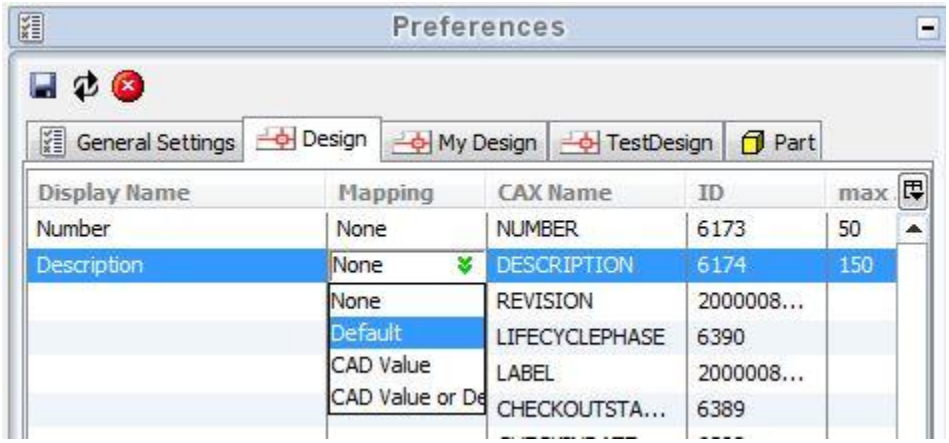
Option	Explanation
Disabled	No viewables are created for drawings.
Enabled	Viewables are created for the selected vieable formats in the *Formats* combo box. If no format is selected, no viewables are creates regardless of the selected option in this combo box.

The following options are available in the *Models* combo box:

Option	Explanation
All Assemblies	Viewables are created only for assembly files, if viewable formats are selected. **Note this option can cause a huge load to process in the CAD session on save and has a big impact on save performance. The option should not be used as a default for all users.**
All Parts	Viewables are created only for part files, if viewable formats are selected. **Note this option can cause a huge load to process in the CAD session on save and has a big impact on save performance. The option should not be used as a default for all users.**
All CAD Files	Viewables are created for all CAD model files, if viewable formats are selected. **Note this option can cause a huge load to process in the CAD session on save and has a big impact on save performance. The option should not be used as a default for all users.**
Top CAD File	Viewables are created only for the root element of a CAD file structure (seen in the tree view), if viewable formats are selected. **Note this option can cause a huge load to process in the CAD session on save and has a big impact on save performance. The option should not be used as a default for all users.**
Disabled	No viewables are created for CAD model files.

Property Value Preferences

Each Design and Part class is represented in the preferences in order to configure the mapping of symbolic CAX properties to fields in PLM. The administrator can setup the mapping interactively. The preferences will be saved into a MCAD-CONFIG filefolder object in PLM if the current user is a member of the admin group. The values have to be set in each subclass independently.



Additionally each field may get a value default mapping.

Property Value Preferences – This section allows you to pre-define the properties that are mapped between CAD and PLM, as part of the save process. By setting these preferences appropriately, you can reduce the use of the interactive save dialog and speed up the save process. The four mapping options are:

- *None* – No value is to be set for this property.
- *Default* – Use the value in the *Default column*.
- *CAD Value* – Use the value defined in the CAD properties, based upon the mapping defined by your administrator.
- *CAD Value or Default* – Use the value defined in the CAD properties, but if no value exists then use the default value in the *Default column*.

CAXConfig.xml Settings

The **CAXConfig.xml** file controls general and numbering options for load and save. The different sections control the communication between client and server, logic for display in the client, parameters for part families and the numbering schemes and change process.

Basic Section

```
<Aliasname>BasicCAXConfig</Aliasname>
<Import>
        <Parameter>
                <FieldCollection>
                        <Field><Name>SITE</Name><Value>MCAD</Value></Field>
                        <Field><Name>FIELD_FOR_NUMBER</Name><Value>DESCRIPTION</Value></Field>
                        <Field><Name>TRANSFER</Name><Value>MULTITHREADED</Value></Field>
                        <Field><Name>OVERWRITE</Name><Value>FALSE</Value></Field>
                        <Field><Name>PRIVILEGES</Name><Value>FALSE</Value></Field>
                        <Field><Name>checkRequired</Name><Value>FALSE</Value></Field>
                        <Field><Name>NonExistingPartsFromCAD</Name><Value>allowed</Value></Field>
                </FieldCollection>
```

Setting	Purpose and available values
SITE	Contains the name of the site to which the client should belong. This can be any string desired without spaces, the default value is `MCAD`. The setting is important for the *Create New* function, the mapping and field configuration (Attributes.xml). See chapter *CAD Startparts Administration in PLM* for details.
FIELD_FOR_NUMBER	**Do not change!**
TRANSFER	MULTITHREADED – simultaneous upload of files SINGLETHREADED – sequential upload of files
OVERWRITE	FALSE – **Do not change!**
PRIVILEGES	FALSE – disable privilege detection to increase performance on LDAP authenticated systems TRUE – enable privilege detection to show up in save preview
checkRequired	FALSE – disable check of required fiels

	TRUE – enable check of required fields
NonExistingPartsFromCAD	remove – will remove non-existing Item numbers from the save preview, if the given Item number does not exist in PLM (Item.NUMBER field is checked) allow – allows CAD mapping to create new Items using the Item.NUMBER field
AllowManualItemNumber	true – make Item Number editable in Details Dialog false –Item Number is not editable in Details Dialog
use.ecx.web.service	true – communicate with Oracle Engineering Collaboration Web Services**. Do not change or remove this setting!**
UseGroovy	true – enable customizable dialogs false – don't enable groovy interpreter (default)

ConnectionProperties Section

```
<Structure>
        <Name>ConnectionProperties</Name>
        <FieldCollection>
                <Field><Name>timeout</Name><Value>900000</Value></Field>
                <Field><Name>agile.selector</Name><Value>Agile</Value></Field>
                <Field><Name>agile.extensions</Name><Value>/integration/services</Value></Field>
                <Field><Name>agile.coreservices</Name><Value>/CoreService/services</Value></Field>
                <Field><Name>attributes.location</Name><Value>use_internal\Attributes.xml</Value></Field>
                <Field><Name>bulksize</Name><Value>25</Value></Field>
                <Field><Name>resolution_max_toplevels</Name><Value>5</Value></Field>
                <Field><Name>resolution_max_depth</Name><Value>2</Value></Field>
                <Field><Name>compression</Name><Value>false</Value></Field>
                <Field><Name>upload</Name><Value>asynchron</Value></Field>
                <Field><Name>handling</Name><Value>asynchron</Value></Field>
                <Field><Name>thumbnail</Name><Value>false</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
timeout	Timeout in milliseconds for webservice requests **Default**: **900000**
newFMS	false – for use with 9.3.1 and 9.3.0 releases (Default) true – for use with 9.3.2 releases
skip-empty	true – removes empty fields from transfer in webservice calls (Default) false – don't remove empty fields from webservice calls
agile.selector	The Agile PLM application identifier. **Default**: **Agile**
agile.extensions	The URL path appendix to your server name where the webservice extensions are located **Default**: **/integration/services**
agile.coreservices	The URL path appendix to your server name where the PLM Core services like BusinessObject are located **Default**: **/CoreService/services**
agile.designsearch	Customized URL appendix to your preferred search mask **Default: /object/search/Basic/Designs/Design**

attributes.location	Path to a local Attributes.xml master file to override the internal file. **Do not change!** **Default**: **use_internal**
bulksize	The server calls are divided into packages of this number of objects per call. Depending on the server and network performance you can increase or decrease the value **Default**: **25**
resolution_max_toplevels	On Load to CAD the structure resolution is called with this maximum number of top elements. If there are more top elements given, the structure resolution is called multiple times in packages to reduce the server load **Default**: **5**
resolution_max_depth	On Load to CAD the structure resolution retrieves only this number of levels per call to reduce the server load. Incomplete nodes are detected and the structure resolution runs recursively for incomplete nodes Default: **2**
compression	**Do not change!** **Default: false**
upload	asynchron – upload files and free CAD session during upload and PLM commit synchron – free CAD session after file upload and PLM commit. **Default**: **asynchron**
handling	asynchron – upload files while CAD action is in progress and free CAD session during upload and PLM commit synchron – upload files after CAD action is complete. **Default**: **asynchron**
thumbnail	true – trigger assembly thumbnail regeneration on save false – no thumbnail regeneration on save. **Default**: **false**

BrowserDisplay Section

```
<Structure>
        <Name>BrowserDisplay</Name>
        <FieldCollection>
                <Field><Name>HideExternalReferences</Name><Value>0</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
HideExternalReferences	External References may lead to a confusing tree view. By default external references are not displayed for more than 50 objects, which can lead to disconnected nodes in the tree view. The number of objects when the external references are suppressed is now configurable in CAXConfig.xml. If the value is set to 0 (zero) external references are never suppressed. **Default**: **0**
HideSuccessSummary	true – if no error occurred then the success message at the end of the PLM commit is not shown. false – (Default)
LazyLoad	true – Load PLM data in the save preview for modified files and on demand/select events only (Default) false – Load PLM data for all files immediately on startup of the save preview
Tree.Menu.Zip	true – show the “Zip and upload workspace” menu item false - suppress the “Zip and upload workspace” menu item (Default)
FindJSessionID	true – try to retrieve a JSessionID from the server (not recommended) false - suppress retrieval of the JSessionID from the server (Default)
SearchNumberCase	1 – convert Number to upper case on search in Details view 2- convert Number to lower case on search in Details view 0 – don’t convert Number on search in Details view

	Default: **0**
LoadWhereUsed	true – enable selective where used context menu in Load Preview false – disable selective where used context menu in Load Preview (Default) **Default: false**

TableDisplay Section

The TableDisplay section contains all columns that should be displayed in the tables of the *Save* and *Load Preview* windows. The fields initially listed here should not be deleted, this may cause errors in program execution. However, it is possible to add additional PLM fields to customize the view. To add an additional column to the table a new field must be added to the FieldCollection of the TableDisplay section.

A field of a PLM Design object is added by insertion of a new line of XML code that can be derived from the following example:

```
<Field><Name>FieldID</Name><Value>0</Value></Field>
```

If a PLM field of a Part object should be used, the string `Item.`**FieldID** must be used as field name instead of just **FieldID**:

```
<Field><Name>Item.FieldID</Name><Value>0</Value></Field>
```

The name of the field must contain the **FieldID** (internal ID) of the PLM field that should be added, its value must always be `0` (zero) by convention. The **FieldID**, also called Base ID, can be found in the Agile Java Client (*Data Settings* → *Classes* → Doubleclick on a Design or Part class → *User Interface Tabs* → Doubleclick on a list entry → *Attributes* tab → Column: *Base ID*).

After having added one or more column entries the file **%USERHOME%\GUIConfig.xml** must be deleted in order to make the changes appear in the GUI.

The following code example shows the TableDisplay section with two customized columns. To the tables the PLM fields 1305 of the Design object and 1313 of the Part object were added as separate columns:

```
<Structure>
        <Name>TableDisplay</Name>
        <!-- Fields available in Save and Load dialog lists -->
        <FieldCollection>
                <Field><Name>CAX_FIL_NAME</Name><Value>0</Value></Field>
                <Field><Name>CAX_FULL_NAME</Name><Value>0</Value></Field>
                <Field><Name>COMPONENTTYPE</Name><Value>0</Value></Field>
                <Field><Name>NUMBER</Name><Value>0</Value></Field>
                <Field><Name>REV</Name><Value>0</Value></Field>
                <Field><Name>REVISION</Name><Value>0</Value></Field>
                <Field><Name>DESCRIPTION</Name><Value>0</Value></Field>
                <Field><Name>LABEL</Name><Value>0</Value></Field>
                <Field><Name>LIFECYCLEPHASE</Name><Value>0</Value></Field>
                <Field><Name>CHECKOUTUSER</Name><Value>0</Value></Field>
                <Field><Name>WORKFLOW_STATUS</Name><Value>0</Value></Field>
                <Field><Name>FILE_STATUS</Name><Value>0</Value></Field>
                <Field><Name>GET</Name><Value>0</Value></Field>
                <Field><Name>CAX_MODIFIED</Name><Value>0</Value></Field>
                <Field><Name>PLM_MODIFIED</Name><Value>0</Value></Field>
                <Field><Name>SAVE_OPTION</Name><Value>0</Value></Field>
                <Field><Name>SAVED</Name><Value>0</Value></Field>
                <Field><Name>HAS_PRIVILEGE</Name><Value>0</Value></Field>
                <Field><Name>FILTER</Name><Value>0</Value></Field>
                <Field><Name>ASSIGNED</Name><Value>0</Value></Field>
                <Field><Name>Item.NUMBER</Name><Value>0</Value></Field>
                <Field><Name>Item.REV</Name><Value>0</Value></Field>
                <Field><Name>Item.ECO</Name><Value>0</Value></Field>
                <Field><Name>Item.LIFECYCLEPHASE</Name><Value>0</Value></Field>
                <Field><Name>Item.DESCRIPTION</Name><Value>0</Value></Field>
                <Field><Name>CAX_MODEL_TYPE</Name><Value>0</Value></Field>
                <Field><Name>CAX_MODEL_REF</Name><Value>0</Value></Field>
                <Field><Name>CAX_LINK_TYPE</Name><Value>0</Value></Field>
                <Field><Name>CAX_LINK_REF</Name><Value>0</Value></Field>
                <Field><Name>CAX_TYPE</Name><Value>0</Value></Field>
                <!-- Customized fields: -->
                <Field><Name>1305</Name><Value>0</Value></Field>
                <Field><Name>Item.1313</Name><Value>0</Value></Field>
        </FieldCollection>
</Structure>
```

DateFormats Section

This section configures the date parser for fields mapped from CAD into PLM. The CAD formats transferred from CAD have to be configured here to enable proper transfer into Agile PLM date fields. Multiple formats are allowed. The formats are specified in the Java Simple Date Format syntax.

```
<Structure>
        <Name>DateFormats</Name>
        <FieldCollection>
                <Field><Name>yyyy-MM-dd HH:mm:ss z</Name><Value>CAD</Value></Field>
                <Field><Name>yyyy-MM-dd</Name><Value>CAD</Value></Field>
                <Field><Name>dd.MM.yyyy HH:mm:ss z</Name><Value>CAD</Value></Field>
        </FieldCollection>
</Structure>
```

FieldConfiguration Section

```
<Structure>
        <Name>FieldConfiguration</Name>
        <FieldCollection>
                <Field><Name>BOM:CAX-IDENT</Name><Value>2175</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
BOM:CAX-IDENT	Agile PLM Field ID for BOM functions for CAD Ident. This option controls direct BOM transfer, but not BOM's published from the CAD Design structure. **Default**: **2175**

WorkspaceTableDisplay Section

The WorkspaceTableDisplay section contains information on the columns of the table displayed in the *Workspace Manager* window. It can be manipulated in the same way as the TableDisplay section. Refer to the chapter *TableDisplay Section* for details, all information given in this chapter apply for the WorkspaceTableDisplay section as well.

CAD_SYSTEMS and CAD_FILE_EXTENSIONS Section

```
<Structure>
    <Name>CAD_SYSTEMS</Name>
    <FieldCollection>
        <!-- Value: CAD files can have copies/backups (Yes/No)-->
        <Field><Name>Catia</Name><Value>No</Value></Field>
        <Field><Name>ProE</Name><Value>Yes</Value></Field>
        <Field><Name>NX</Name><Value>No</Value></Field>
        <Field><Name>SolidEdge</Name><Value>No</Value></Field>
        <Field><Name>SolidWorks</Name><Value>No</Value></Field>
        <Field><Name>Inventor</Name><Value>No</Value></Field>
    </FieldCollection>
</Structure>
```

```
<Structure>
    <Name>CAD_FILE_EXTENSIONS</Name>
    <FieldCollection>
        <Field><Name>CATProduct</Name><Value>Catia</Value></Field>
        <Field><Name>CATPart</Name><Value>Catia</Value></Field>
        <Field><Name>CATDrawing</Name><Value>Catia</Value></Field>
        <Field><Name>prt</Name><Value>ProE</Value></Field>
        <Field><Name>asm</Name><Value>ProE</Value></Field>
        <Field><Name>drw</Name><Value>ProE</Value></Field>
```

The CAD_SYSTEMS section defines the list of supported CAD Tools.

Setting	Purpose and available values
CAD Tool Name	Yes – there are multiple file versions per object No – there is only one CAD file per object

The entries are required for handling of file versions like in Pro/E and to define the valid file extensions in the `CAD_FILE_EXTENSIONS` section. Only files listed in the `CAD_FILE_EXTENSIONS` section are displayed in the *Workspace Manager*.

Processes Section

```
<Structure>
        <Name>Processes</Name>
        <FieldCollection>
                <Field><Name>cnext</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>xtop</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>ugraf</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>edge</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>sldworks</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>inventor.exe</Name><Value>OnlyNamesUsed</Value></Field>
                <Field><Name>XPlmBulkLoaderTool.exe</Name><Value>OnlyNamesUsed</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
Process name list	Process names are listed here. If none of the listed processes is running then the EC Web Connector and Workspace Manager is shut down automatically.

ThreadPool Section

```
<Structure>
        <Name>ThreadPool</Name>
        <FieldCollection>
                <Field><Name>NumberOfThread</Name><Value>10</Value></Field>
                <Field><Name>Timeout</Name><Value>1800000</Value></Field>
                <Field><Name>MaxRetryNumber</Name><Value>3</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
NumberOfThread	Number of simultaneous file uploads and downloads. **Default**: **10**
Timeout	Timeout for file operations in milliseconds. **Default**: **1800000**
MaxRetryNumber	If an upload or download fails, the system retries the operation this number of times. If the operation still fails, an error is thrown. **Default**: **3**

OverrideConfiguration Section

```
<Structure>
        <Name>OverrideConfiguration</Name>
        <FieldCollection>
                <Field><Name>DisplayedDesignClasses</Name><Value>All</Value></Field>
                <Field><Name>DisplayedItemClasses</Name><Value>All</Value></Field>
                <Field><Name>DisplayedChangeClasses</Name><Value>All</Value></Field>
                <Field><Name>DisplayedDesignAutonumbers</Name><Value>All</Value></Field>
                <Field><Name>DisplayedItemAutonumbers</Name><Value>All</Value></Field>
                <Field><Name>DisplayedChangeAutonumbers</Name><Value>All</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
DisplayedDesignClasses	List of Design classes that display in UI **Default**: **All**
DisplayedItemClasses	List of Item classes that display in UI **Default**: **All**
DisplayedChangeClasses	List of Change classes that display in UI

	Default: **All**
DisplayedDesignAutonumber	List of Design autonumbers that display in UI **Default**: **All**
DisplayedItemAutonumber	List of Item autonumbers that display in UI **Default**: **All**
DisplayedChangeAutonumber	List of Change autonumbers that display in UI **Default**: **All**

Viewables Section

```
<Structure>
        <Name>Viewables</Name>
        <FieldCollection>
                <Field><Name>ViewablesDrawing</Name><Value>PDF;TIF</Value></Field>
                <Field><Name>ViewablesModel</Name><Value>STEP;IGES;JT;X_T</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
ViewablesDrawing	List of viewable types for drawings available in UI **Default**: **PDF;TIF**
ViewablesModel	List of viewable types for drawings available in UI **Default**: **STEP;IGES;JT**

PartFamilies Section

Setting	Purpose and available values
FamilySelection	true – on selection of a generic or instance always all instances and the generic is selected all together false – don't handle all in one **Default**: **false**
FamilyInstanceNumbering	GENERIC_INDEX – Instances and configurations get the same number as the generic, plus a counter. GENERIC_CONFIG – Instances and configurations get the same number as the generic, plus the configuration name. False – Instances and configurations get independent numbers from the generic. **Default**: **GENERIC_INDEX**
CountDelimiter	Separator char between the number and the counter for drawings an part family members **Default**: _
PartInstanceDashNumbering	true – the Part item corresponding to part family instance get the suffix from the design instance (like P001-001,…) false - the Part item corresponding to part family instances get own item numbers, no instance suffix. **(Default)**
PartGenericHasDashNumber	true – the Part item corresponding to the generic Design has a dash number (like P001-000, instance will be P001-001,…) false - the Part item corresponding to the generic Design has no dash number (like P001, instance will be P001-001,…)
PartInstanceDelimiter	Separator char between the number and the counter for part variants **Default**: **-**

ChangeProperties Section

```
<Structure>
        <Name>ChangeProperties</Name>
        <FieldCollection>
                <Field><Name>InitialRevision</Name><Value>-</Value></Field>
                <Field><Name>RevisionSequence</Name>
                 <Value>-,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z</Value></Field>
                <Field><Name>RevisionSequenceEditor</Name><Value>true</Value></Field>
                <Field><Name>SetLifecyclePhase</Name><Value>true</Value></Field>
                <!-- DesignRevisionLogic: increment=increase minor number on checkout, publish=set
                major rev from attached part, editable=editable from save and CAD,
                noparentheses=remove parentheses from revision -->
                <Field><Name>DesignRevisionLogic</Name>
                 <Value>increment,editable,publish,noparentheses</Value></Field>
                <Field><Name>InitialDesignRevision</Name><Value>-</Value></Field>
                <Field><Name>VersionSeparator</Name><Value>.</Value></Field>
                <Field><Name>InitialVersion</Name><Value>1</Value></Field>
                <Field><Name>InitialVersionCheckin</Name><Value>checkin</Value></Field>
                <Field><Name>InitialPublishVersion</Name><Value>1</Value></Field>
                <Field><Name>PartDesignNumbering</Name><Value>false</Value></Field>
                <Field><Name>PushPartRevisionToDesign</Name><Value>true</Value></Field>
                <Field><Name>PushDesignRevisionToPart</Name><Value>true</Value></Field>
                <Field><Name>PublishAttachments</Name><Value>any</Value></Field>
                <Field><Name>PublishAttachmentType</Name><Value>NONE</Value></Field>
                <Field><Name>PublishIntroductory</Name><Value>true</Value></Field>
                <!-- Site to use when publishing Item/BOM, NONE to switch it off: -->
                <Field><Name>PublishPartSite</Name><Value>none</Value></Field>
        </FieldCollection>
</Structure>
```

Setting	Purpose and available values
InitialRevision	Initial Item revision value
RevisionSequence	Comma separated list of valid revision codes for Items
RevisionSequenceEditor	true – enable the editor in the save preview table false – disable the editor **Default**: **true**
SetLifecyclePhase	true - transfer and set the Item lifecycle phase false – don't set the Item lifecycle phase **Default**: **true**
DesignRevisionLogic	Comma separated list of the following values. If set the values control the following behavior: increment – increase the minor revision code in the design revision field. editable – the Design revision field is editable in the save preview. publish – the Design revision is reset on publish to fit the item revision

	code. The minor revision is calculated or reset. noparentheses – remove any parentheses from the design revision **Default**: **increment,editable,publish,noparentheses**
InitialDesignRevision	Initial Design revision value
VersionSeparator	Separator char between the major and minor design revision code. **Default**: **. (Dot)**
InitialVersion	Initial Version number **Default**: **1**
InitialVersionCheckin	checkin – checkin the initial Design version increment – checkin the initial Design version and checkout again to keep reservation. false – no automated checkin of initial Design versions **Default**: **checkin**
InitialPublishVersion	Initial Version number on publish **Default**: **1**
PartDesignNumbering	false – Item and Design have different number sources and numbers don't have to match between the part and the design. true – Designs use Part autonumbers and the Design number matches the Part number. **Default**: **false**
PushPartRevisionToDesign	true – On editing the part revision editor in the save preview the part revision is set to the Design major revision. false – no synchronization on edit **Default**: **true**
PushDesignRevisionToPart	true – On editing the design revision in the save preview the major design revision is set to the Part revision. false – no synchronization on edit

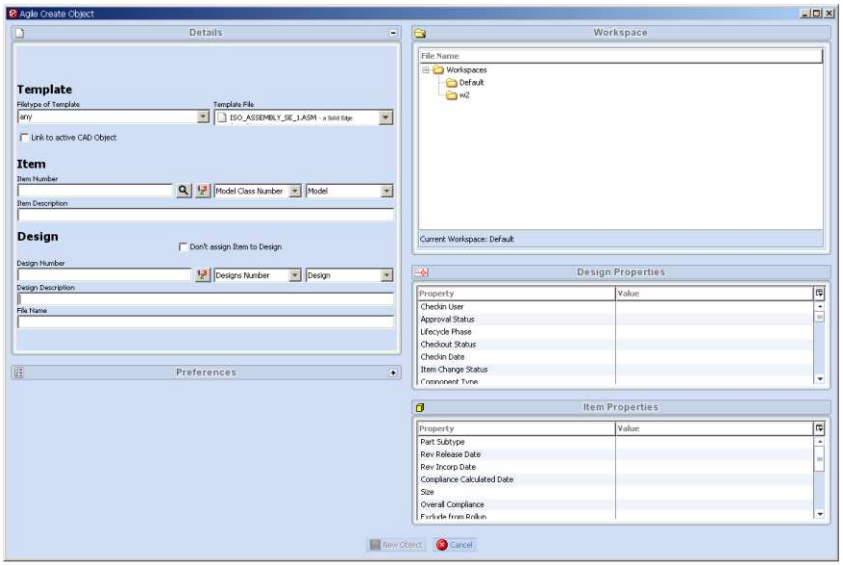
	Default: **true**
PublishAttachments	Comma separated list of file extensions that will be attached to the Item during the publish step. **Default**: **any**
PublishAttachmentType	Value to be set to the Item Attachments in the Attachment Type field for files attached by EC. **Note that the value must be contained in the available values list for the attachment type field. Use Agile administration to edit the list.** **Default**: **NONE**
PublishIntroductory	true – publish to Introductory Item revisions with no change object is allowed. false – don't publish to Introductory Item revisions **Default**: **true**
PublishPartSite	none – no specific manufacturing site is set to the Item on publish Any other value is interpreted as the desired Site name to be set to the Item on publish **Default**: **none**
MaxDesignNumberLength	Maximum length of number fields in PLM, default is 50 **Default**: **50**
AllowedNumberCharacters	set to allowed character set for number parsing if special characters should be removed **Default**: **ANY**
Change_Category	Default Change Category for generated changes Deafult: none
PublishErrorHandling	WARNING – publish errors are reported as warnings (Default) ERROR – publish errors are reported to the user as errors
PushPartLifecycleToDesign	true – set the part lifecycle phase to the design object on save.

	false – don't synchronize the part lifecycle to the Design object **Default**: **false**
PublishIntegerQuantities	true – BOM quantities are set as integers like 1, 2, 5 false – BOM quantities are set as doubles like 1.0, 2.0, 5.0 **Default**: **false**
PublishToReleasedItems	true – enables publishing to released items. Note that the user needs special privileges in Agile PLM to perform this successful false – disable publishsing to released items completely even if Agile PLM privileges would allow it. **Default**: **false**

CAD Startparts Administration in PLM

This section provides detailed information about all necessary steps to enable object creation with the Create Object dialog. The dialog can be accessed from the *New* entry of the *Agile* submenu in the menu bar/ribbon bar of user's CAD system. Once the dialog was opened the user can choose a template file and a Design number, based on these two parameters a new object can be created simultaneously in CAD and PLM.

Figure: Create Object frame, used to create new objects



Creating the Template Structure in Agile

The Create Object dialog uses a certain data tructure in the Agile PLM as the basis of object creation. All templates that should be available to the user must be stored in that structure. The structure must be named `%SITE%-START-%CADSYSTEM%` where `%SITE%` is a variable that can be defined in the **CAXConfig.xml** file, its default value is `MCAD` if it is not defined in that file. `%CADSYSTEM%` is the name of the CAD in upper case for which the template structure should be used, the following values are valid:

```
AUTOCAD
CATIA
INVENTOR
NX
PROE
SOLIDEDGE
SOLIDWORKS
```

or

MSOFFICE in case the connector for Microsoft Office is used which is treated as a CAD in this case.

It is possible to store template structures of more than one CAD in a PLM. If the %SITE% variable is assigned for each client, subdivisions of the same CAD can use templates in the same PLM without interfering each other. E. g. LONDON-START-CATIA and ROME-START-CATIA may exist in the same PLM for CATIA clients with the different %SITE% variables LONDON and ROME where the LONDON division can only use the structure LONDON-START-CATIA and the other way round. The %SITE% variable can be assigned by editing the file **<Install Directory>\ini\CAXConfig.xml**. **<Install Directory>** is the client's EC Connector installation folder. The set the value, after the beginning of the first occurrence of the line <FieldCollection> edit the following line:

<Field><Name>SITE</Name><Value>*Custom_site_value*</Value></Field>

Custom_site_value is the string that defines the site, here you can enter any string desired, but it must not contain spaces. Save the **CAXConfig.xml** after finishing the edit. These steps must be repeated for each client that should belong to a site.

To create the template structure, administration rights for the PLM that is desired to be used are necessary. In addition it is necessary to switch off forced autonumber usage when creating new Design objects. This can be achieved with the following steps:

- Login to Agile9 Admin Client
- In the tab pane on the left hand side choose the *Admin* tab
- In the tree in that tab open the node *Data Settings*
- Choose sub-node *Classes* by double-click, a new window opens
- In the new window select the class *Design* by double-click, a new window opens
- In the *General Information* tab set the option *Autonumber Required* to *No*
- Click *Save*
- Close the Admin Client

To create the template structure login to the Agile9 Web Client as a user with appropriate rights to create new Design objects. After the login, click on the *Create New* drop down list and select *File Folders → Designs*. Having clicked on *Designs* a popup window opens, select the type *Design* and enter the appropriate name of the template structure that was mentioned above, e. g. MCAD-START-CATIA. Click *Save*, the template structure has been created and can be loaded with CAD files now.

Adding Template Files to the Structure

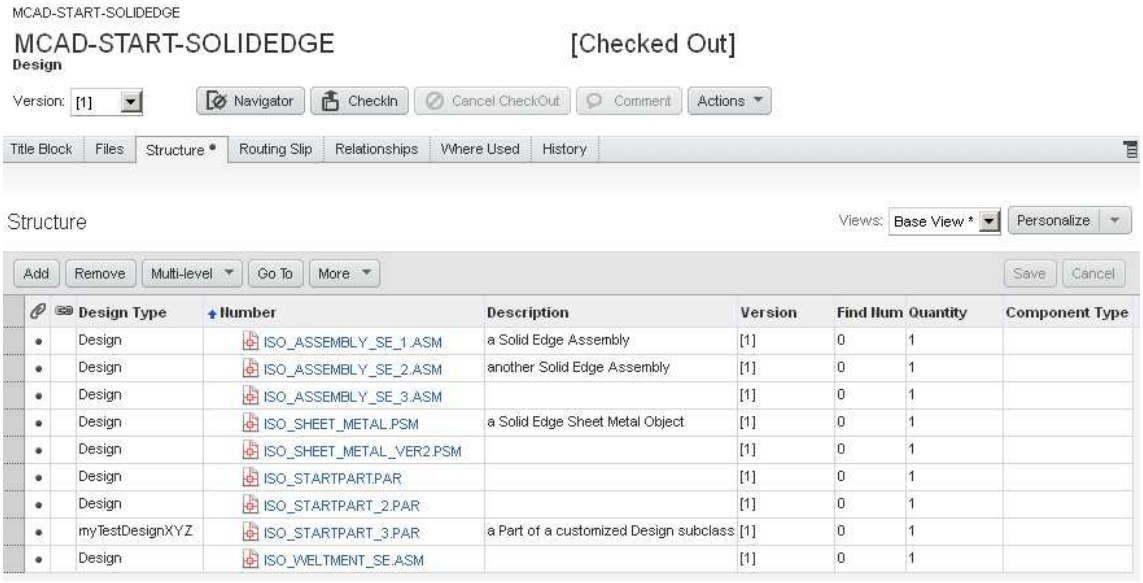
A template file is a normal CAD file that was added to a template structure in the PLM. When a client uses the Create Object frame for object creation, the selected template file is downloaded to the user's workspace, renamed and opened in the user's CAD. Due to this, template files can be created in the same way as any other CAD file.

Loading templates to the PLM is done by adding a new Design object to the *Structure* tab of the template structure. The object can be of any Design subclass. To do so, click on the *Add* button. In the text field that opens click on the button *Create to add*, a popup window opens. In the window select a *Type* and enter a *Number*. Click *Add*, the new Design object for a template file has now been created. After that, select the *Files* tab of the newly created template file object and upload a CAD file (note: the template file object is a Design object in the template structure, not the structure itself which is a Design object itself). That CAD file will become the template file. Having done this, select the *Title Block* tab and enter at least the properties *Design System* and *Filetype*. All other properties can optionally be left empty.

The *Design System* is the name of the CAD from which the template file was created, *Filetype* is the file ending of the template file in upper case (without a dot). Optionally the property *Subtype* can be filled out in order to assign a subtype to the template file. Detailed instructions on how to use subtypes can are given in the chapter *Subtypes* of this documentation.

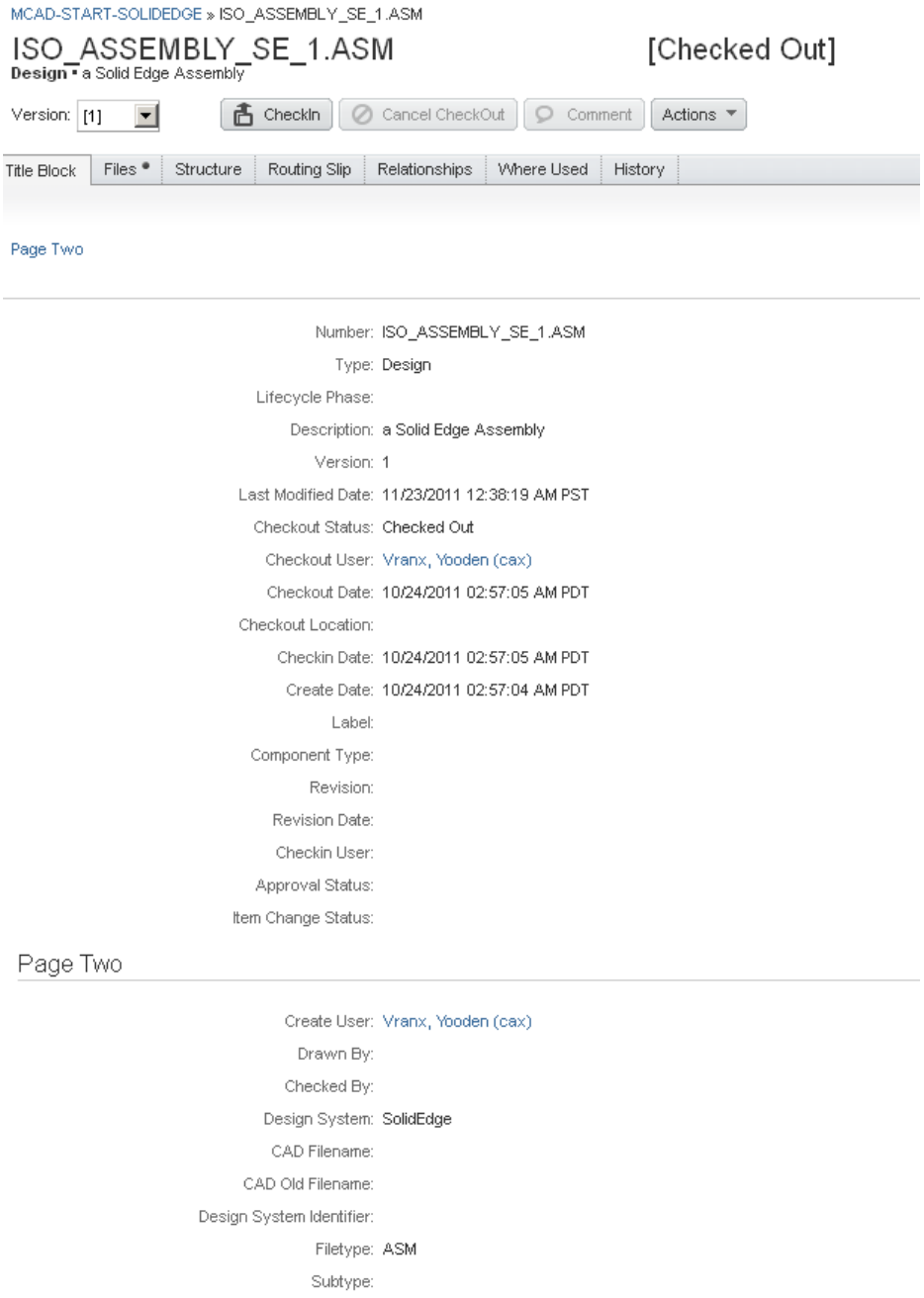
Save the changes that were made to that template object. As soon as this is done the template structure is operational and template creation is possible from the Create Object frame. The following figures show an operational template structure for Solid Edge with several template files attached, any other template structure should look similarly:

Figure: Example template structure for Solid Edge



	Design Type	Number	Description	Version	Find Num	Quantity	Component Type
•	Design	ISO_ASSEMBLY_SE_1.ASM	a Solid Edge Assembly	[1]	0	1	
•	Design	ISO_ASSEMBLY_SE_2.ASM	another Solid Edge Assembly	[1]	0	1	
•	Design	ISO_ASSEMBLY_SE_3.ASM		[1]	0	1	
•	Design	ISO_SHEET_METAL.PSM	a Solid Edge Sheet Metal Object	[1]	0	1	
•	Design	ISO_SHEET_METAL_VER2.PSM		[1]	0	1	
•	Design	ISO_STARTPART.PAR		[1]	0	1	
•	Design	ISO_STARTPART_2.PAR		[1]	0	1	
•	myTestDesignXYZ	ISO_STARTPART_3.PAR	a Part of a customized Design subclass	[1]	0	1	
•	Design	ISO_WELTMENT_SE.ASM		[1]	0	1	

Figure: Example of a Title Block of a template object



Subtypes

A subtype is a CAD file type that extends, or in other words, specializes another CAD file type. Every subtype file has the same file ending as its supertype and acts as a normal CAD file concerning the CAD system. However, in the PLM a subtype behaves slightly different. Every subtype is defined by the string

that is entered in the *Subtype* property of a Design's *Title Block*. If a Design should not be assigned to a subtype, the property field can be left empty. Note: A valid subtype entry must consist of at least two characters (spaces do not count as characters in this case). If only one character is entered in that property field it is treated as if it was empty.

To create a new subtype, you only need to assign a string that was not already used to a Design object in the template structure. It is not necessary to create a new Design object when creating a new subtype, changing or deleting the subtype of an already existing Design is possible as well. To do so, you only need to change or delete the entry of the field *Subtype* in the Design's *Title Block*. Likewise, the subtype of a Design object can be changed to an already existing subtype by simply changing the *Subtype* property. It is possible to assign the same subtype to template objects of different file types as well.

Every subtype that is used in the template structure gets its separate entry in the *Filetype of Template* combo box in the Create Object frame. The entry is displayed the same way as the entry for the supertype with the name of the subtype appended. If a subtype entry is selected, only the template files that belong to the selected file type **and** the selected subtype are displayed in the *Template File* combo box. However, if the entry for the supertype is selected, all template files of the selected file type are displayed, no matter to which subtype they belong. (Note that the entry for a supertype in the Filetype of Template combo box does only appear if there is at least one template file of the corresponding file type that does not belong to a subtype.)

Subtypes enable the administrator to separate template files from each other according to certain criteria. They may be used to provide templates of the same file type to the user in a structured way, e. g. separated by projects, locations, names of clients and so on. In fact, every property that can be expressed as a string could actually be used as a subtype making a finely graded template classification possible.

Possible Errors

This section gives a summary on the most common error messages that might occur during the usage of the Create Object frame. Error messages are usually displayed in a popup window above the main window causing that window not to respond to user interaction as long as the popup is open. The errors are roughly divided into two kinds: Critical messages are caused by errors that do not allow a save continue of program execution, however, the main window is not closed in this case leaving the last parameter settings visible. Info messages are caused by unproper configurations or missing data but they do not cause serious dangers for program execution, anyway, in most cases the user cannot

continue the action that caused the message.

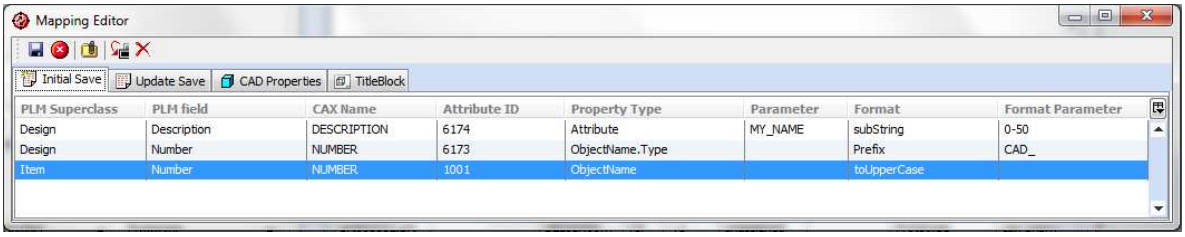
Table: Overview of common errors

Error Message	Explanation
"No templates found in database. New objects cannot be created until templates are created in the database."	An info message, telling the user that the template structure in the PLM is empty or was not found. Due to this, no new objects can be created. **Possible solution:** Make sure that the template section in Agile9 has been properly created and contains files.
"The template download failed. Check if network connection to PLM is correctly established and the PLM is responsive."	A critical error, telling the user that the download of a template file failed. This message usually appears after another error message. **Possible solution:** In most cases this error message appears, when the user's computer lost the network connection to the PLM or the PLM does not response any more.
A template object does not appear in the Template File combo box although it is part of the template structure in PLM.	Not an error message, but damaged template objects in PLM are not displayed in the Create Object frame, neither in the Filetype of Template combo box nor in the Template File combo box. **Possible solution:** Check if all properties of the template object's Title Block are properly set. Refer to the chapters *Adding Template Files to the Structure* and *Subtypes* for details.
Template files that contain family tables do not work	CAD files with family tables of any kind (called iAssemblies and iParts (Inventor), Generics and Instances (Pro Engineer), Configurations (Solid Works), Templates and Members (NX)) should not be used as template files because references between or within these parts cannot be resolved when downloading a template file using the *Create* function. Due to this it is not guaranteed that the family tables work in the downloaded template. **Possible solution:** Do not use family tables in template files.

Note: The error messages listed here are given in English language, error messages may appear in the language of your localization, too.

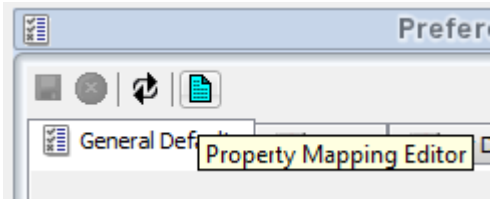
Mapping Editor

This section provides a complete summary of the options that are available in the Mapping Editor. The Mapping Editor is used to define mappings of CAD properties to PLM fields during save. It is used for mapping of PLM values to CAD properties or the drawing titleblock as well.

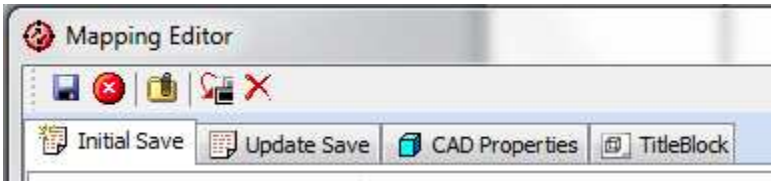


Using the Mapping Editor

The mapping is CAD-specific. For each CAD system, a separate mapping is created. The Mapping Editor is launched using the blue button from the preferences dialog in the save preview. The button is only visible if you have admin privileges in PLM.



The toolbar of the Mapping Editor has the following functionality.



Save the mapping to disk. In order to use or test the new mapping, you have to exit the save preview and launch the save preview again from CAD. The mapping definition is read on each start of a save, update properties or *Update Titleblock* command.

Attention: The local save doesn't make the mapping available to all users and will be lost if you restart the integration. In order to have a permanent mapping, store it to PLM as explained below.

Save and attach the mapping into PLM and make it available to all users. In order to use or test the new mapping, you have to exit the save preview and launch the save preview again from CAD. The mapping definition is read on each start of a save, update properties or *Update Titleblock* command. **All other client machines have to relogin using the "Disconnect Session" command in order to download the updated mapping.**

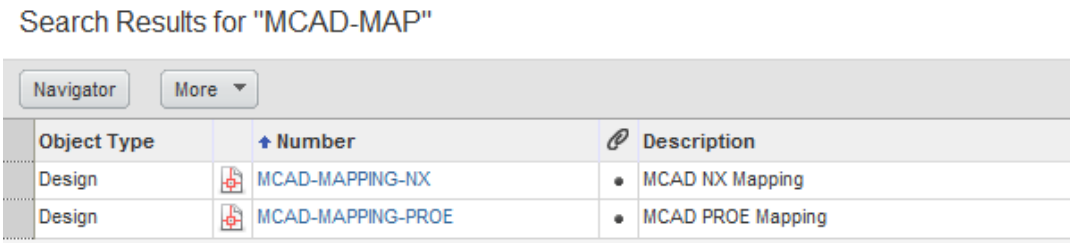
Cancel all changes to the mappings and reread the latest saved mapping.

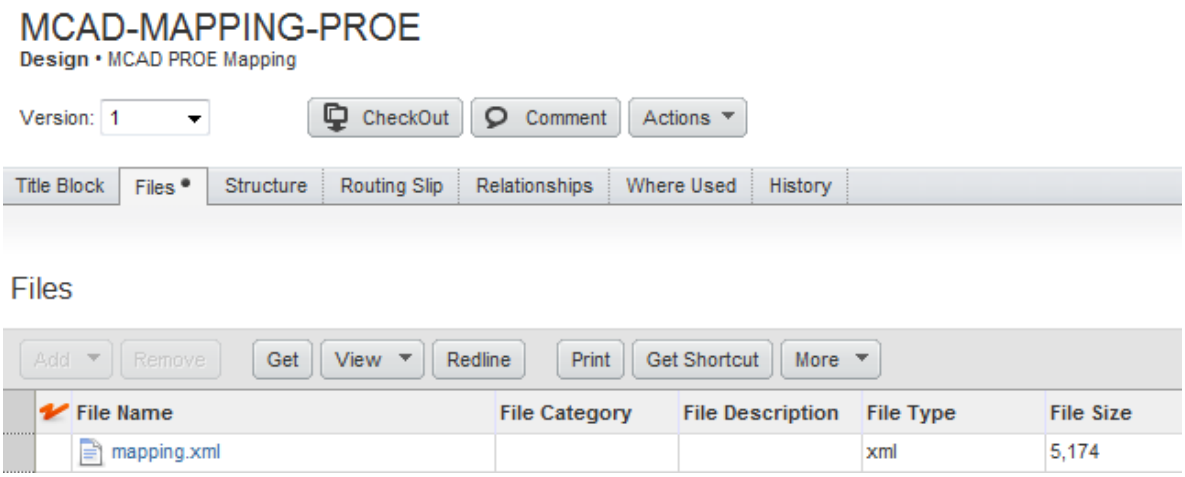
Append a row into the current active mapping tab.

Remove a selected row in the current active mapping tab.

MCAD-MAPPING folders – How the mapping is handled

The system creates a design called `MCAD-MAPPING-%CADSYSTEM%` and attaches the mapping to that Design. On the next login of any user with the same CAD system, the mapping is downloaded automatically and used on the client machine.

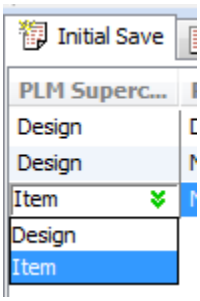

Search Results for "MCAD-MAP"
Navigator
More
Object Type
Number
Description
Design
MCAD-MAPPING-NX
MCAD NX Mapping
Design
MCAD-MAPPING-PROE
MCAD PROE Mapping


MCAD-MAPPING-PROE
Design • MCAD PROE Mapping
Version: 1
CheckOut
Comment
Actions
Title Block
Files
Structure
Routing Slip
Relationships
Where Used
History
Files
Add
Remove
Get
View
Redline
Print
Get Shortcut
More
File Name
File Category
File Description
File Type
File Size
mapping.xml
xml
5,174

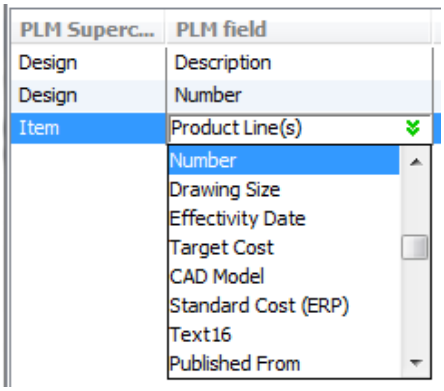
Mapping CAD properties to PLM fields

The first two tabs define the mappings of CAD properties to PLM fields. For CAD objects that aren't known in PLM the mapping in the *Initial Save* tab is used. The *Update Save* tab is used for CAD objects that already have an assigned Agile object. Both sections are configured the same way but may contain different settings. For instance on initial save the predefinition of the Design number or the assigned Item number is important. On update save there is only the need to map attributes like dimensions or descriptions.

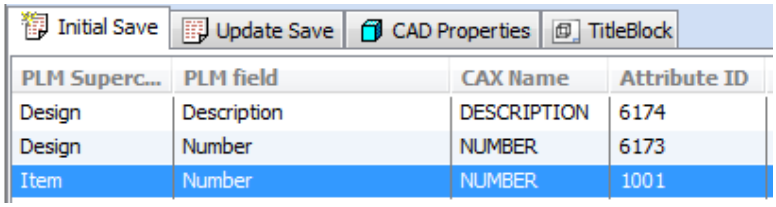
The first column switches the target Agile superclass, which can be *Design* or *Item*.



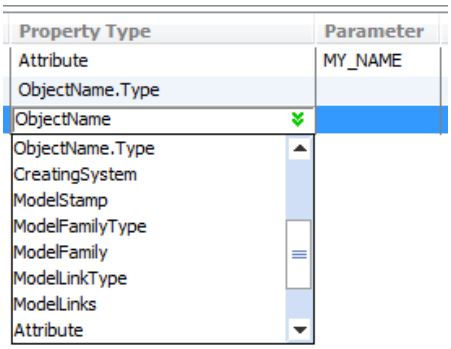
Depending on the selection in the PLM superclass column the available PLM fields are filtered from the current class configuration. Only visible and editable fields in PLM are available.



After selecting the target field, additional information for this field displays in the *CAX Name* and the *Attribute ID* column. Both are read-only.



In the *Property Type* column, you can select from CAD internal integration parameters and from CAD properties. If you set the value to *Attribute* you have to specify a CAD property name in the *Parameter* column.



Note: A parameter with this name is searched in the configuration specific properties first. If there is no configuration specific property with this name, the standard or custom properties of the part are scanned.

The available property types are CAD dependent and listed in the table below.

Property Type	Purpose
ModelFullName	File name including Path

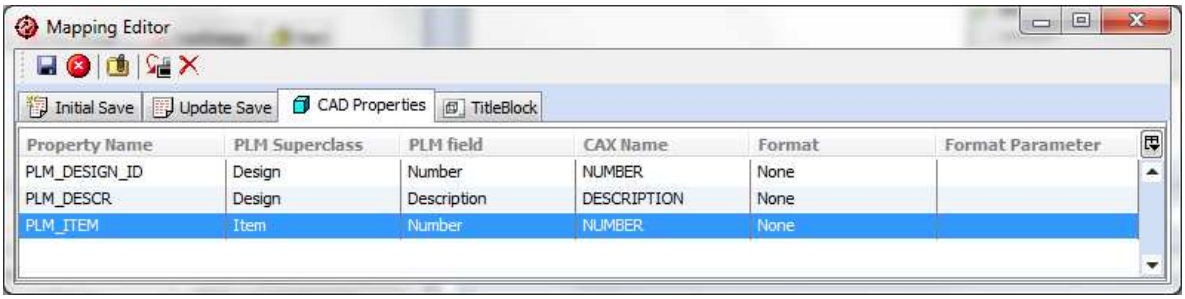
ModelName	File name without Path
ModelExtension	File extension
ModelType	File type, equals extension in most CAD system
ModelPath	File path location
ModelNameOrConfiguration	File name or configuration name
ModelNameAndConfiguration	File name plus configuration name
ObjectName	File name without path and extension
ObjectName.Type	File name without path
CreatingSystem	CAD version the file is created with
Attribute	Retrieve the CAD Property defined in the Parameter column
String	Set the string defined in the Parameter column
Code	Execute the CAD callback code defined in the Parameter column
$USER	Set the current login user name as value
$USERID	Set the current login user ID as value
Configuration	Configuration name
ModelConfigurationNames	Contained configuration names
ModelStamp	Internal timestamp of the file
ModelFamilyType	Part family type or configuration type
ModelFamily	Part family or configuration master or generic
ModelLinkType	Linked references type
ModelLinks	Linked source file
SimplifiedRep	Simplified representation identifier (Pro/E)
DrawingModel	Model assigned to the drawing (Pro/E)
DrawingModelName	Model name assigned to the drawing (Pro/E)

DrawingModelType	Model extension assigned to the drawing (Pro/E)
HelperPartIdent	Helper part ident

Additionally you can now specify a format as described in the chapter *Formatting values during mapping*. Save the mapping as needed locally, and into PLM if you want all engineers to use it.

Mapping PLM values to CAD Properties

The mapping of PLM values back into CAD properties is defined in two sections. The CAD tab defines the mappings to CAD properties. Some CAD systems support special logic for drawing titleblocks, especially if the displayed texts in the drawing cannot be linked to CAD properties. For this use case the second *TitleBlock* tab is used by some CAD tools.



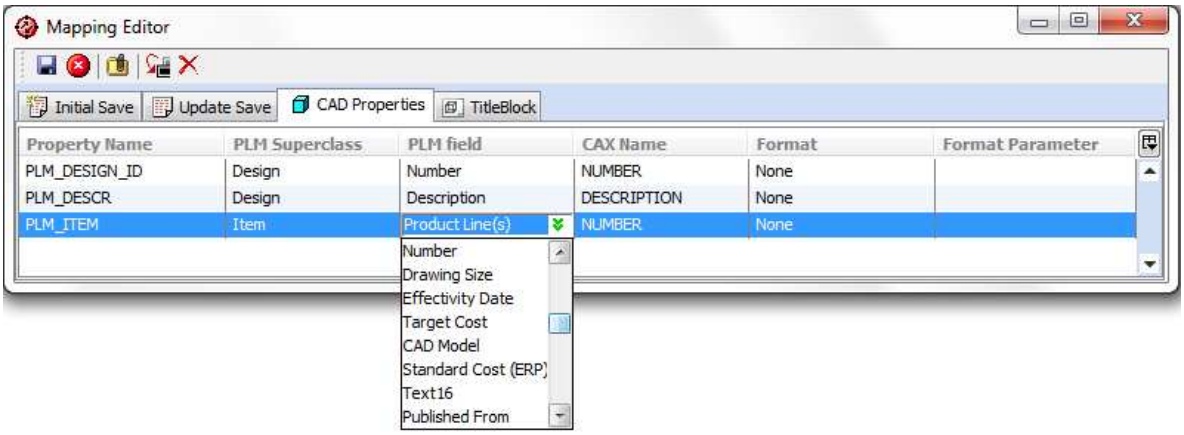
The first column contains the name of the target CAD property. In the CAD tab this is the name of the CAD part attribute, CAD property or configuration-specific property.

Attention: Some CAD Tools use predefined placeholders for some internal CAD property names like listed in the table below.

CAD Tool	Property Name	Purpose
CATIA V5	PartNumber	Internal Part ID

CATIA V5	Nomenclature	Internal Part Description
CATIA V5	Definition	Internal Definition field
CATIA V5	Description	Internal Description field
CATIA V5	DescriptionReference	Internal Reference field
CATIA V5	Revision	Internal Revision ID

In the second column you define the PLM superclass, from which the value should be sent to CAD. You can map values from the Item and the Design object. Once you have selected the desired super class you can choose from the list of available attributes of this class.

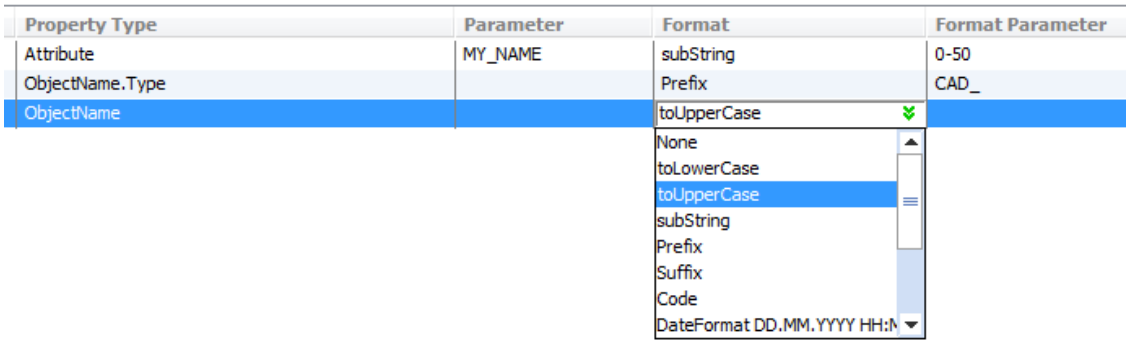


The *CAX Name* column is now set with the default symbolic name or the attribute ID if no such symbolic name exists. Note that the *CAX Name* column is editable to support editing complex legacy logic for data extraction in drawing titleblocks.

Additionally, you can now specify a format as described in the chapter *Formatting values during mapping*. Save the mapping as needed locally and into PLM if you want all engineers to be using it.

Formatting values during mapping

The *Format* and *Format Parameter* columns provide basic formatting options for values mapped between CAD and PLM and vice versa.



Valid format options are listed in the table below. The date formats work only if the value to be formatted is given in an integer value.

Format	Purpose
None	No formatting action is executed
toLowerCase	Convert the value to lower case characters
toUpperCase	Convert the value to upper case characters
subString	Cut a substring from the value with the start and end index defined in the Format Parameter column. Valid values are for example 0-end, 3-end, 0-50.
Prefix	Append the prefix defined in the Format Parameter column in front of the value
Suffix	Append the suffix defined in the Format Parameter column at the end of the value
Code	Execute the CAD callback code defined in the Format Parameter column to format the value

DateFormat DD.MM.YYYY HH:MI:SS	Format the Date like 15.12.2010 23:30:00
DateFormat DD.MM.YYYY	Format the Date like 15.12.2010
DateFormat DD.MM.YY	Format the Date like 15.12.10
DateFormat DD-MM-YY	Format the Date like 15-12-10
DateFormat MM/DD/YY	Format the Date like 12/15/10
DateFormat DD-MMM-YY	Format the Date like 15-Dec-2010

Language and Localization Administration

This section provides information about how to setup the GUI languages for the integration. Languages are set up on three different components. The CAD connector, EC Web Components and the PLM user language. All are independent from each other.

PLM User and Data Language

The preferred user language controlled the data values which are displayed in EC dialogs and transferred between CAD and PLM. The setup is done in the user preferences settings in PLM.

EC Web Components

The EC Web Component dialogs are localized. The desired language on runtime is defined using a switch in **(components)\com\acx.bat**. Valid values are `EN`, `FR`, `DE`.

```
start /b javaw.exe %JAVA_HEAP_SIZE% -Dcom.xplm.agile.Language=EN -Djava….
```

Use the disconnect session command or restart the CAD Tool after changing the setting.

CAD Connector Components

Due to different techniques for integration in CAD tools, the Addin in CAD (CAD menus and icons) must be configured CAD-specific.

CAD Tool	How to configure Addin Language
SolidWorks	The CAD Addin language is configured in **components\xml\xPLMConnector.xml** in this setting: `<Language>EN</Language>`. Valid values are `EN`, `FR`, `DE`.
Solid Edge	The CAD Addin language is configured in **components/xml/xPLMConnector.xml** in this setting: `<Language>EN</Language>`. Valid values are `EN`, `FR`, `DE`.
Pro/ENGINEER and Creo	The CAD Addin language is configured in **xACP.cfg** with the `AcpLang=English` setting. Valid values are `English`, `French`, `German`.

CATIA V5	The CATIA CAD system language controls the language of EC menus and toolbars

Agile PLM Server Class Administration

This section provides a complete summary of class configuration options required for the EC Web connector. Once the basic installation has been done following the instructions in the Installation Guide, you can refer here for details of all required settings.

The EC Web Connector requires an essential minimum set of fields enabled to work properly. Make sure all the fields listed in the following chapters are visible and enabled on your Agile server and all PLM users that should work with the EC . You may need to enable additional fields according to the desired CAD property mapping.

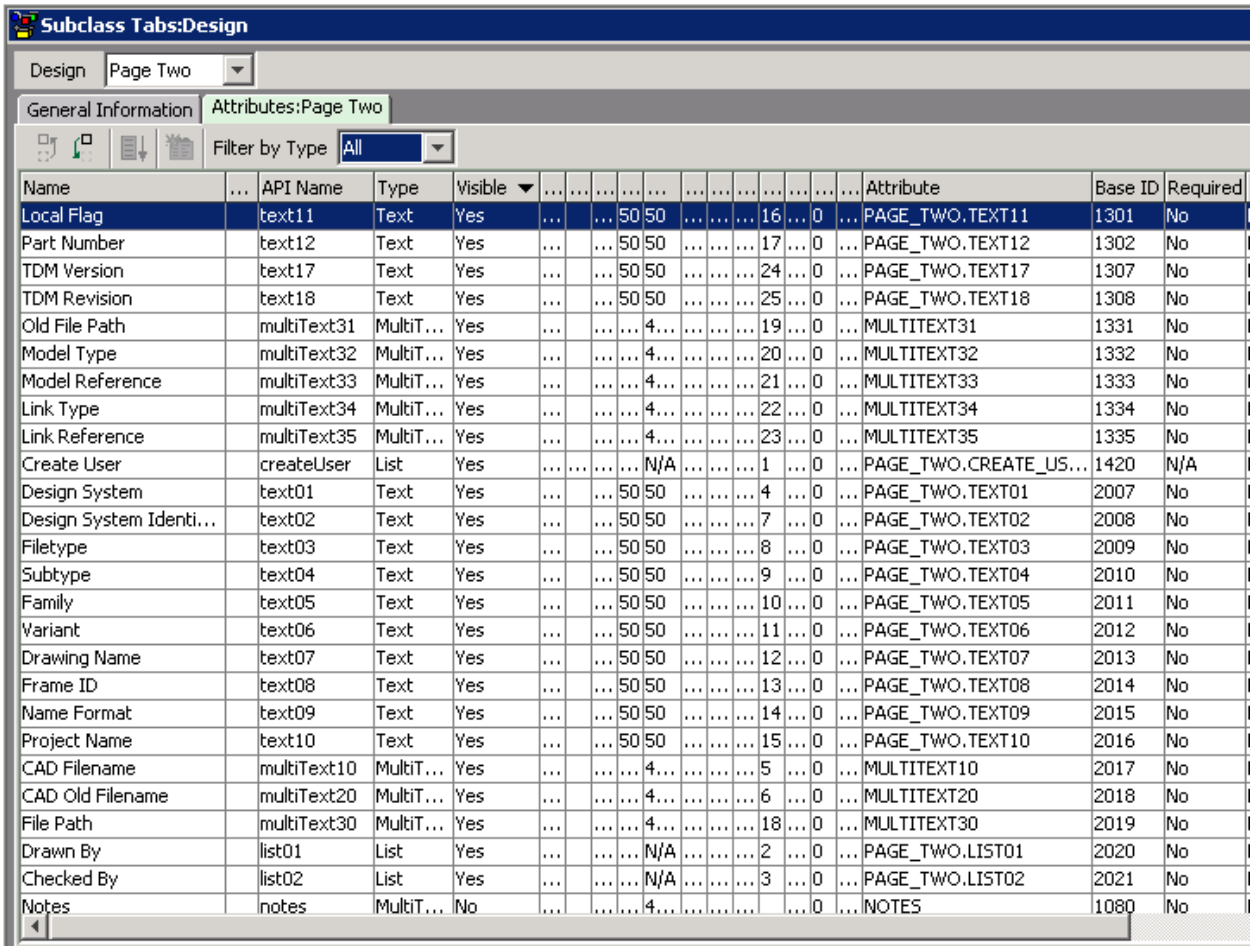
Note that in most cases, the attribute name is predefined, although it may be disabled by default. Make sure these are all *enabled*. Attributes where the name is not predefined are mentioned below. **Please also make sure the User Roles and Modify Privileges are setup for the login user to discover, modify and read all of the fields in the following tables.**

Designs – Page Two

Make sure the following fields are enabled:

- Local Flag (ID: 1301)
- Part Number (ID: 1302)
- Model Type (ID: 1332)
- Model Reference (ID: 1333)
- Link Type (ID: 1334)
- Link Reference (ID: 1335)
- Design System (ID: 2007)
- Design System Identifier (ID: 2008)
- Filetype (ID: 2009)
- Subtype (ID: 2010)
- Family (ID: 2011)
- Variant (ID: 2012)
- Drawing Name (ID: 2013)
- Frame ID (ID: 2014)
- Name Format (ID: 2015)

- Project Name (ID: 2016)
- CAD Filename (ID: 2017)
- File Path (ID: 2019)



Name	API Name	Type	Visible	Attribute	Base ID	Required
Local Flag	text11	Text	Yes	PAGE_TWO.TEXT11	1301	No
Part Number	text12	Text	Yes	PAGE_TWO.TEXT12	1302	No
TDM Version	text17	Text	Yes	PAGE_TWO.TEXT17	1307	No
TDM Revision	text18	Text	Yes	PAGE_TWO.TEXT18	1308	No
Old File Path	multiText31	MultiT...	Yes	MULTITEXT31	1331	No
Model Type	multiText32	MultiT...	Yes	MULTITEXT32	1332	No
Model Reference	multiText33	MultiT...	Yes	MULTITEXT33	1333	No
Link Type	multiText34	MultiT...	Yes	MULTITEXT34	1334	No
Link Reference	multiText35	MultiT...	Yes	MULTITEXT35	1335	No
Create User	createUser	List	Yes	PAGE_TWO.CREATE_US...	1420	N/A
Design System	text01	Text	Yes	PAGE_TWO.TEXT01	2007	No
Design System Identi...	text02	Text	Yes	PAGE_TWO.TEXT02	2008	No
Filetype	text03	Text	Yes	PAGE_TWO.TEXT03	2009	No
Subtype	text04	Text	Yes	PAGE_TWO.TEXT04	2010	No
Family	text05	Text	Yes	PAGE_TWO.TEXT05	2011	No
Variant	text06	Text	Yes	PAGE_TWO.TEXT06	2012	No
Drawing Name	text07	Text	Yes	PAGE_TWO.TEXT07	2013	No
Frame ID	text08	Text	Yes	PAGE_TWO.TEXT08	2014	No
Name Format	text09	Text	Yes	PAGE_TWO.TEXT09	2015	No
Project Name	text10	Text	Yes	PAGE_TWO.TEXT10	2016	No
CAD Filename	multiText10	MultiT...	Yes	MULTITEXT10	2017	No
CAD Old Filename	multiText20	MultiT...	Yes	MULTITEXT20	2018	No
File Path	multiText30	MultiT...	Yes	MULTITEXT30	2019	No
Drawn By	list01	List	Yes	PAGE_TWO.LIST01	2020	No
Checked By	list02	List	Yes	PAGE_TWO.LIST02	2021	No
Notes	notes	MultiT...	No	NOTES	1080	No

Designs – Files

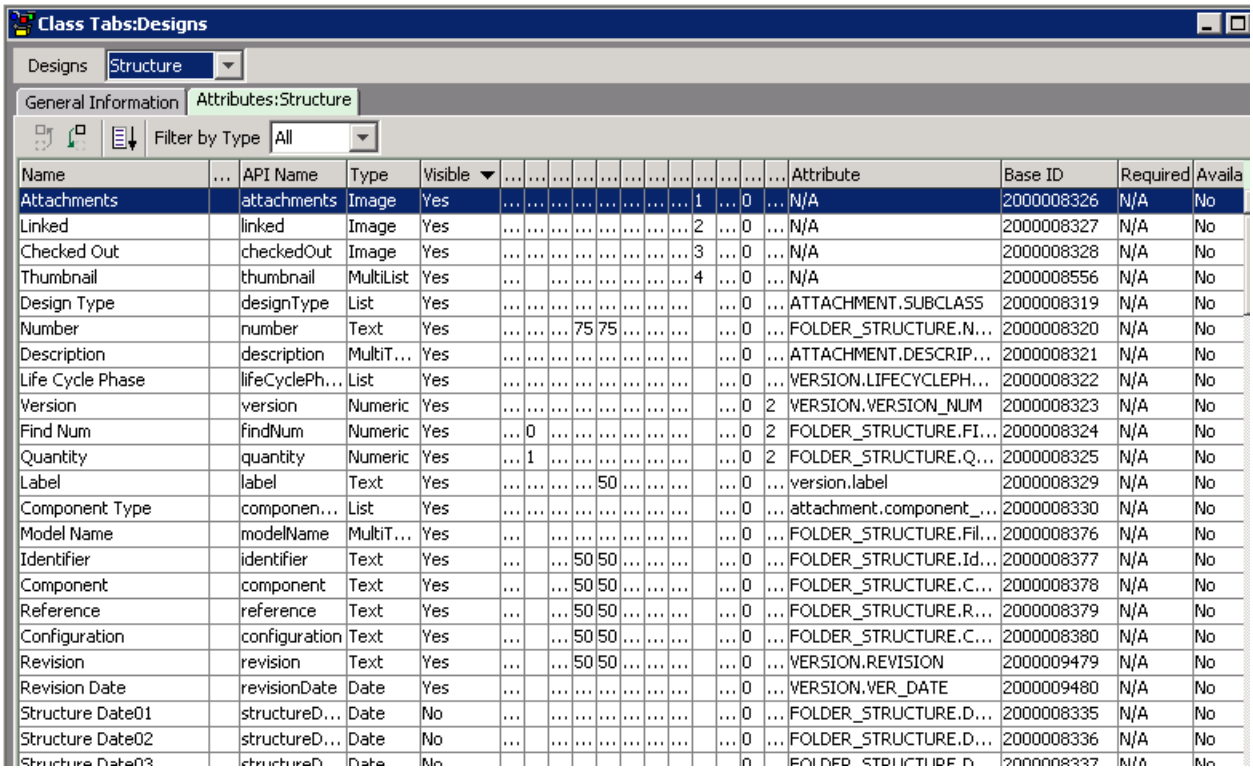
Make sure the following fields are enabled:

- File Category (ID: 2000008509) with a list of available values containing the *Source* and *Viewable* values.

Designs – Structure

Make sure the following fields are enabled:

- Quantity (ID: 2000008325)
- Component Type (ID: 2000008330)
- Model Name (ID: 2000008376)
- Identifier (ID: 2000008377)
- Component (ID: 2000008378)
- Reference (ID: 2000008379)
- Configuration (ID: 2000008380)

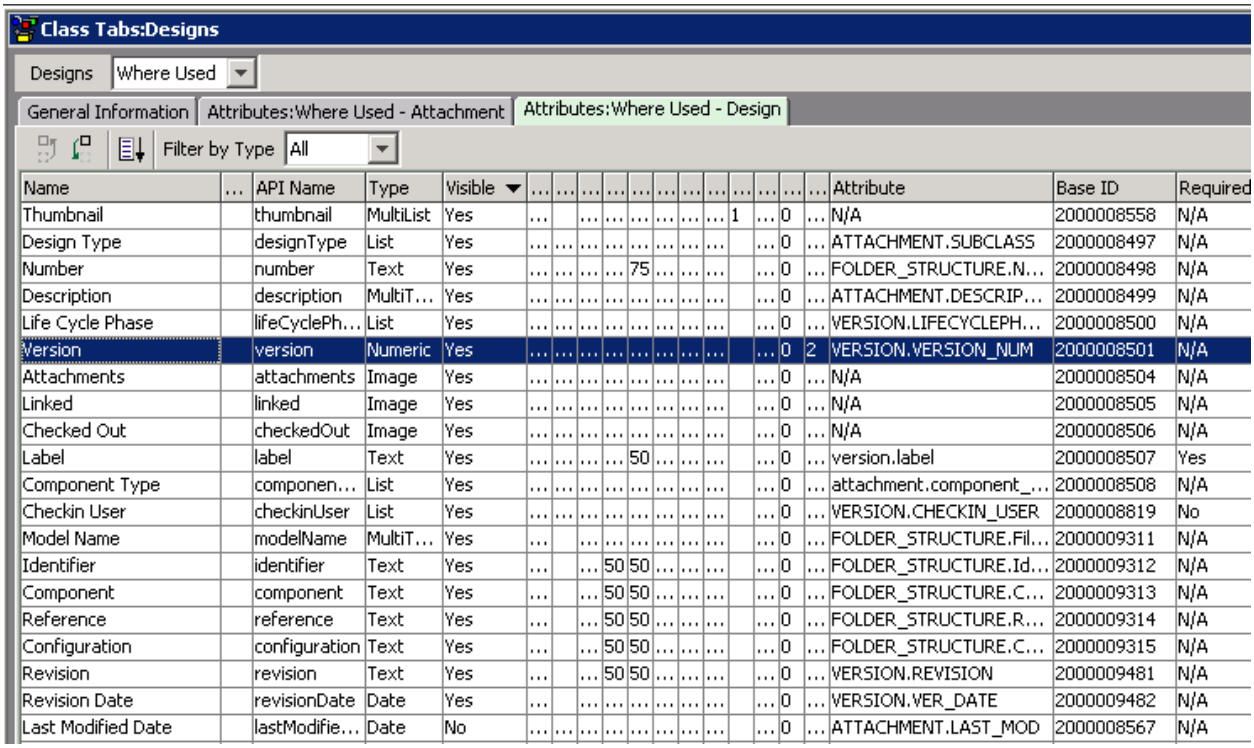


Name	API Name	Type	Visible	Attribute	Base ID	Required	Availa
Attachments	attachments	Image	Yes	N/A	2000008326	N/A	No
Linked	linked	Image	Yes	N/A	2000008327	N/A	No
Checked Out	checkedOut	Image	Yes	N/A	2000008328	N/A	No
Thumbnail	thumbnail	MultiList	Yes	N/A	2000008556	N/A	No
Design Type	designType	List	Yes	ATTACHMENT.SUBCLASS	2000008319	N/A	No
Number	number	Text	Yes	FOLDER_STRUCTURE.N...	2000008320	N/A	No
Description	description	MultiT...	Yes	ATTACHMENT.DESCRIP...	2000008321	N/A	No
Life Cycle Phase	lifeCyclePh...	List	Yes	VERSION.LIFECYCLEPH...	2000008322	N/A	No
Version	version	Numeric	Yes	VERSION.VERSION_NUM	2000008323	N/A	No
Find Num	findNum	Numeric	Yes	FOLDER_STRUCTURE.FI...	2000008324	N/A	No
Quantity	quantity	Numeric	Yes	FOLDER_STRUCTURE.Q...	2000008325	N/A	No
Label	label	Text	Yes	version.label	2000008329	N/A	No
Component Type	componen...	List	Yes	attachment.component_...	2000008330	N/A	No
Model Name	modelName	MultiT...	Yes	FOLDER_STRUCTURE.Fil...	2000008376	N/A	No
Identifier	identifier	Text	Yes	FOLDER_STRUCTURE.Id...	2000008377	N/A	No
Component	component	Text	Yes	FOLDER_STRUCTURE.C...	2000008378	N/A	No
Reference	reference	Text	Yes	FOLDER_STRUCTURE.R...	2000008379	N/A	No
Configuration	configuration	Text	Yes	FOLDER_STRUCTURE.C...	2000008380	N/A	No
Revision	revision	Text	Yes	VERSION.REVISION	2000009479	N/A	No
Revision Date	revisionDate	Date	Yes	VERSION.VER_DATE	2000009480	N/A	No
Structure Date01	structureD...	Date	No	FOLDER_STRUCTURE.D...	2000008335	N/A	No
Structure Date02	structureD...	Date	No	FOLDER_STRUCTURE.D...	2000008336	N/A	No

Designs – Where Used – Design

Make sure the following fields are enabled:

- Version (ID: 2000008501)
- Component Type (ID: 2000008508)
- Model Name (ID: 2000009311)
- Identifier (ID: 2000009312)
- Component (ID: 2000009313)
- Reference (ID: 2000009314)
- Configuration (ID: 2000009315)

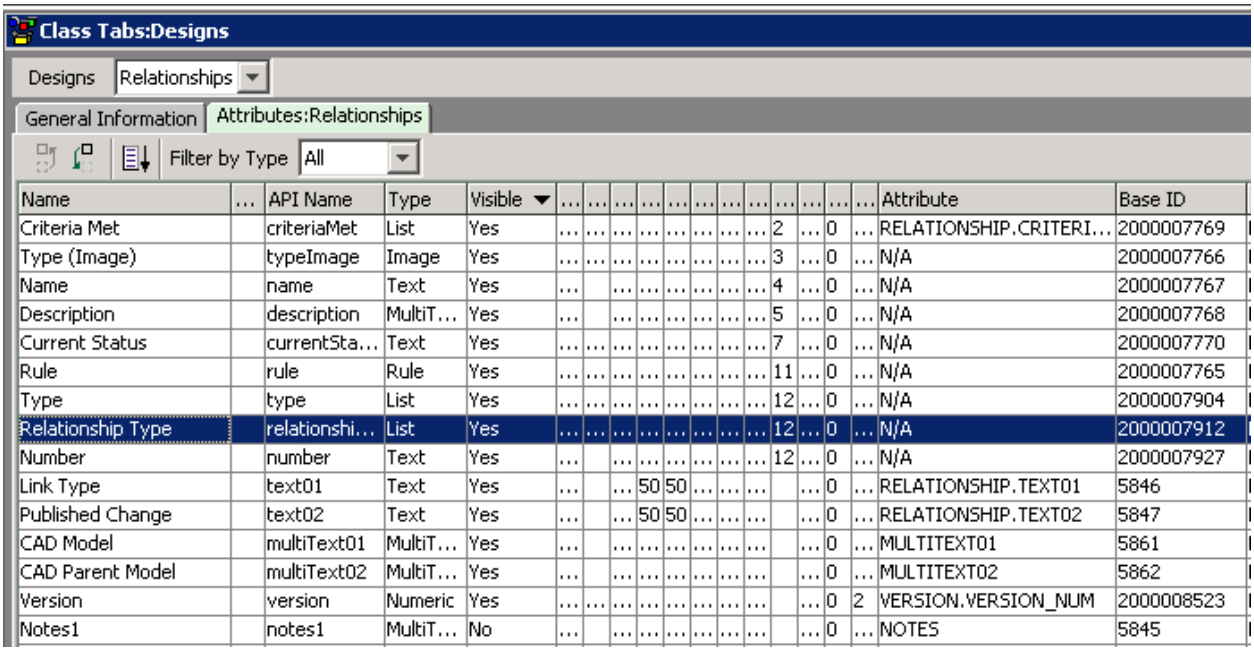


Name	API Name	Type	Visible	Attribute	Base ID	Required
Thumbnail	thumbnail	MultiList	Yes	N/A	2000008558	N/A
Design Type	designType	List	Yes	ATTACHMENT.SUBCLASS	2000008497	N/A
Number	number	Text	Yes	FOLDER_STRUCTURE.N...	2000008498	N/A
Description	description	MultiT...	Yes	ATTACHMENT.DESCRIP...	2000008499	N/A
Life Cycle Phase	lifeCyclePh...	List	Yes	VERSION.LIFECYCLEPH...	2000008500	N/A
Version	version	Numeric	Yes	VERSION.VERSION_NUM	2000008501	N/A
Attachments	attachments	Image	Yes	N/A	2000008504	N/A
Linked	linked	Image	Yes	N/A	2000008505	N/A
Checked Out	checkedOut	Image	Yes	N/A	2000008506	N/A
Label	label	Text	Yes	version.label	2000008507	Yes
Component Type	componen...	List	Yes	attachment.component_...	2000008508	N/A
Checkin User	checkinUser	List	Yes	VERSION.CHECKIN_USER	2000008819	No
Model Name	modelName	MultiT...	Yes	FOLDER_STRUCTURE.Fil...	2000009311	N/A
Identifier	identifier	Text	Yes	FOLDER_STRUCTURE.Id...	2000009312	N/A
Component	component	Text	Yes	FOLDER_STRUCTURE.C...	2000009313	N/A
Reference	reference	Text	Yes	FOLDER_STRUCTURE.R...	2000009314	N/A
Configuration	configuration	Text	Yes	FOLDER_STRUCTURE.C...	2000009315	N/A
Revision	revision	Text	Yes	VERSION.REVISION	2000009481	N/A
Revision Date	revisionDate	Date	Yes	VERSION.VER_DATE	2000009482	N/A
Last Modified Date	lastModifie...	Date	No	ATTACHMENT.LAST_MOD	2000008567	N/A

Designs – Relationships

Make sure the following fields are enabled:

- Link Type (ID: 5846) – not the standard attribute name, must be modified manually
- Published Change (ID: 5847) – not the standard attribute name, must be modified manually
- CAD Model (ID: 5861) – not the standard attribute name, must be modified manually
- CAD Parent Model (ID: 5862) – not the standard attribute name, must be modified manually
- Relationship Type (ID: 2000007912)
- Number (ID: 2000007927)
- Version (ID: 2000008523)

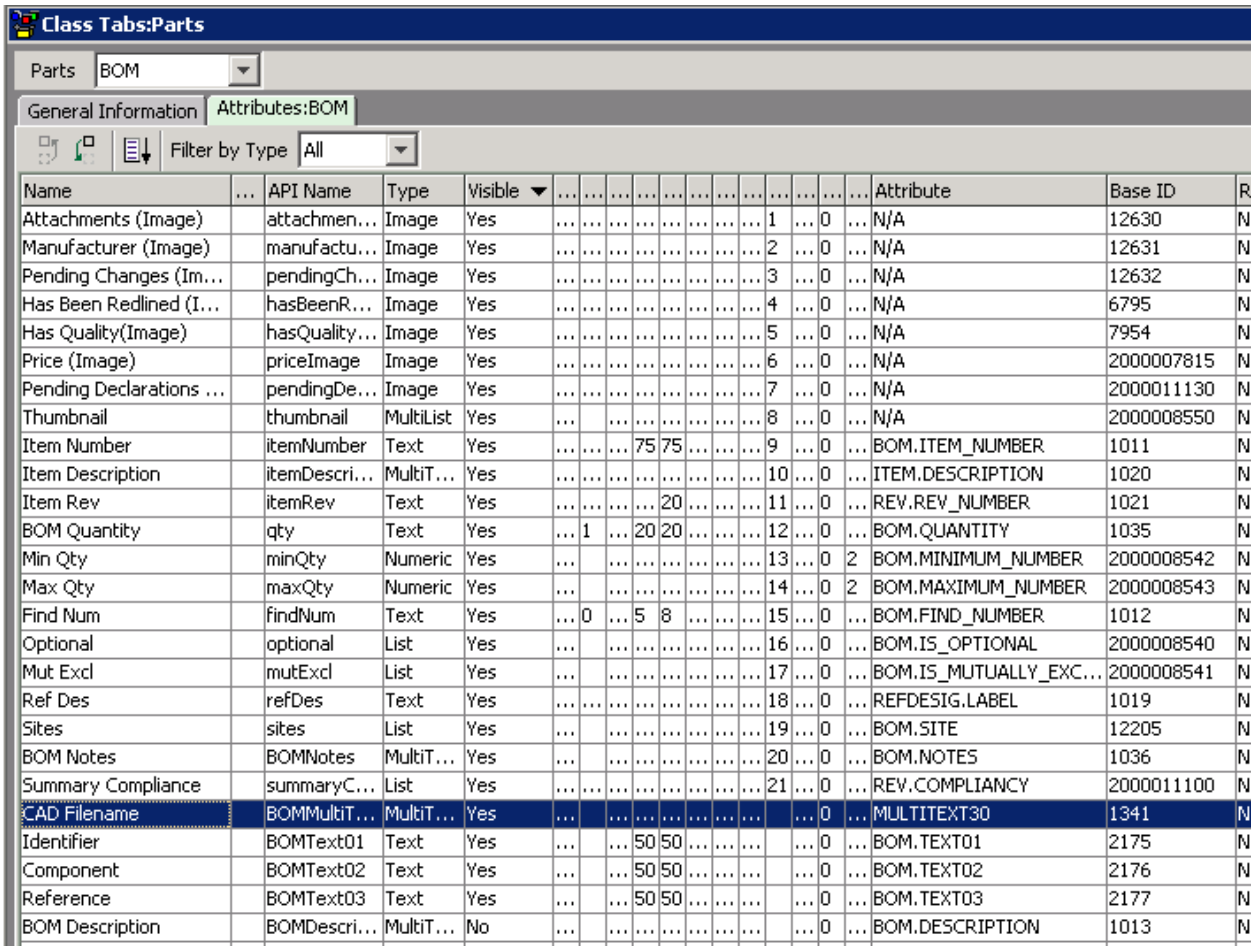


Name	API Name	Type	Visible	Attribute	Base ID
Criteria Met	criteriaMet	List	Yes	RELATIONSHIP.CRITERI...	2000007769
Type (Image)	typeImage	Image	Yes	N/A	2000007766
Name	name	Text	Yes	N/A	2000007767
Description	description	MultiT...	Yes	N/A	2000007768
Current Status	currentSta...	Text	Yes	N/A	2000007770
Rule	rule	Rule	Yes	N/A	2000007765
Type	type	List	Yes	N/A	2000007904
Relationship Type	relationshi...	List	Yes	N/A	2000007912
Number	number	Text	Yes	N/A	2000007927
Link Type	text01	Text	Yes	RELATIONSHIP.TEXT01	5846
Published Change	text02	Text	Yes	RELATIONSHIP.TEXT02	5847
CAD Model	multiText01	MultiT...	Yes	MULTITEXT01	5861
CAD Parent Model	multiText02	MultiT...	Yes	MULTITEXT02	5862
Version	version	Numeric	Yes	VERSION.VERSION_NUM	2000008523
Notes1	notes1	MultiT...	No	NOTES	5845

Parts – BOM

Make sure the following fields are enabled:

- BOM Quantity (ID: 1035)
- CAD Filename (ID: 1341) – not the standard attribute name, must be modified manually
- Identifier (ID:2175) – not the standard attribute name, must be modified manually
- Component (ID: 2176) – not the standard attribute name, must be modified manually
- Reference (ID:2177) – not the standard attribute name, must be modified manually

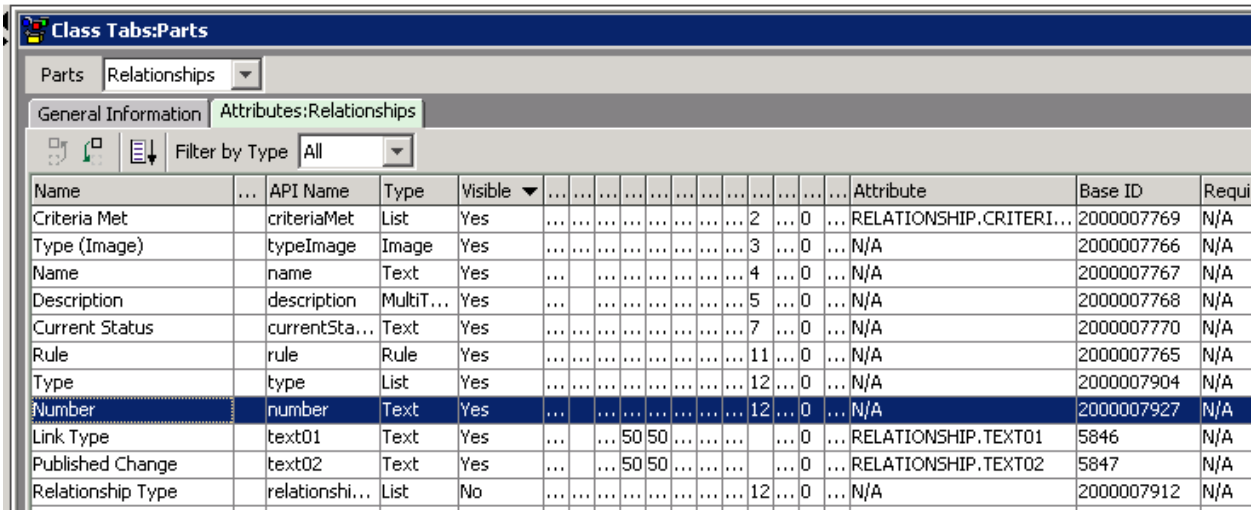


Name	API Name	Type	Visible	Attribute	Base ID
Attachments (Image)	attachmen...	Image	Yes	N/A	12630
Manufacturer (Image)	manufactu...	Image	Yes	N/A	12631
Pending Changes (Im...	pendingCh...	Image	Yes	N/A	12632
Has Been Redlined (I...	hasBeenR...	Image	Yes	N/A	6795
Has Quality(Image)	hasQuality...	Image	Yes	N/A	7954
Price (Image)	priceImage	Image	Yes	N/A	2000007815
Pending Declarations ...	pendingDe...	Image	Yes	N/A	2000011130
Thumbnail	thumbnail	MultiList	Yes	N/A	2000008550
Item Number	itemNumber	Text	Yes	BOM.ITEM_NUMBER	1011
Item Description	itemDescri...	MultiT...	Yes	ITEM.DESCRIPTION	1020
Item Rev	itemRev	Text	Yes	REV.REV_NUMBER	1021
BOM Quantity	qty	Text	Yes	BOM.QUANTITY	1035
Min Qty	minQty	Numeric	Yes	BOM.MINIMUM_NUMBER	2000008542
Max Qty	maxQty	Numeric	Yes	BOM.MAXIMUM_NUMBER	2000008543
Find Num	findNum	Text	Yes	BOM.FIND_NUMBER	1012
Optional	optional	List	Yes	BOM.IS_OPTIONAL	2000008540
Mut Excl	mutExcl	List	Yes	BOM.IS_MUTUALLY_EXC...	2000008541
Ref Des	refDes	Text	Yes	REFDESIG.LABEL	1019
Sites	sites	List	Yes	BOM.SITE	12205
BOM Notes	BOMNotes	MultiT...	Yes	BOM.NOTES	1036
Summary Compliance	summaryC...	List	Yes	REV.COMPLIANCY	2000011100
CAD Filename	BOMMultiT...	MultiT...	Yes	MULTITEXT30	1341
Identifier	BOMText01	Text	Yes	BOM.TEXT01	2175
Component	BOMText02	Text	Yes	BOM.TEXT02	2176
Reference	BOMText03	Text	Yes	BOM.TEXT03	2177
BOM Description	BOMDescri...	MultiT...	No	BOM.DESCRIPTION	1013

Parts – Relationships

Make sure the following fields are enabled:

- Link Type (ID: 5846) – not the standard attribute name, must be modified manually
- Published Change (ID: 5847) – not the standard attribute name, must be modified manually
- Number (ID: 2000007927)



Name	API Name	Type	Visible	Attribute	Base ID	Requi
Criteria Met	criteriaMet	List	Yes	RELATIONSHIP.CRITERI...	2000007769	N/A
Type (Image)	typeImage	Image	Yes	N/A	2000007766	N/A
Name	name	Text	Yes	N/A	2000007767	N/A
Description	description	MultiT...	Yes	N/A	2000007768	N/A
Current Status	currentSta...	Text	Yes	N/A	2000007770	N/A
Rule	rule	Rule	Yes	N/A	2000007765	N/A
Type	type	List	Yes	N/A	2000007904	N/A
Number	number	Text	Yes	N/A	2000007927	N/A
Link Type	text01	Text	Yes	RELATIONSHIP.TEXT01	5846	N/A
Published Change	text02	Text	Yes	RELATIONSHIP.TEXT02	5847	N/A
Relationship Type	relationshi...	List	No	N/A	2000007912	N/A

Parts – Pending Changes

Make sure the following fields are enabled:

- New Lifecycle Phase
- Workflow
- Change Category

Configuring Engineering Collaboration Clients for HTTPS

Introduction

If the MCAD connectors are utilizing the Agile PLM Core and EC Services need to use the HTTPS protocol instead of the HTTP protocol, some additional work and configuration needs to be done.

This description assumes that the server side (either the application server itself or an HTTPS proxy server) is already configured to run with HTTPS. This should be validated with the help of the Agile web client.

For making the client side properly working with the HTTPS server, the server's Certificate Authority (CA) certificate needs to be imported into a Java Keystore (JKS) for the clients and a client certificate will need to be generated with the help of this CA certificate.

This document only covers the so-called Mutual Authentication (two-way authentication) for SSL, which is considered the most secure one.

Creating Client Keystore for Mutual Authentication

When wanting to use HTTPS with the Agile PLM application server, there should already be a server JKS file available. Please check the application server's documentation on where to find it. If there is such a file already, please skip step 1 (Create the keystore for the server) and start with step 2. Please do only step 1 if you would like to create self-signed certificates for testing purposes. Self-signed certificates should not be used in production environments.

Any data entered in the samples below is just for demonstration purposes and must be replaced with actual values.

The Java Keystore (JKS) files are named "plm-server.jks" and "plm-client.jks" in the samples below. It could be replaced by any other name in all the command lines.

The examples below use the password "Agile123", but it is advised to use a password that is not that easy to guess. When you enter it, please do not type the square brackets.

Create the keystore for the server

When there is no JKS file available on the server, create a keystore for the server by executing following command in a terminal. "plm-server" in the following command corresponds to the private key/self signed public key certificate alias in the keystore while "plm-server.jks" is the name of the creating keystore file.

```
# keytool -genkey -alias plm-server -keyalg RSA -keystore plm-server.jks

Enter keystore password:  Agile123

Re-enter new password: Agile123

What is your first and last name?

  [Unknown]:  My Company

What is the name of your organizational unit?

  [Unknown]:  PLM

What is the name of your organization?

  [Unknown]:  My Company

What is the name of your City or Locality?

  [Unknown]:  Everywhere

What is the name of your State or Province?

  [Unknown]:  World Wide

What is the two-letter country code for this unit?

  [Unknown]:  WW

Is CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW correct?

  [no]:  yes


Enter key password for <plm-server>

        (RETURN if same as keystore password):
```

Create the keystore for the client

As we need to have a JKS file for the client in any case, create a client keystore named "plm-client.jks" with the alias "plm-client" using following command.

```
# keytool -genkey -alias plm-client -keyalg RSA -keystore plm-client.jks

Enter keystore password:  Agile123

Re-enter new password: Agile123

What is your first and last name?

  [Unknown]:  My Company

What is the name of your organizational unit?

  [Unknown]:  PLM

What is the name of your organization?

  [Unknown]:  My Company

What is the name of your City or Locality?

  [Unknown]:  Everywhere

What is the name of your State or Province?

  [Unknown]:  World Wide

What is the two-letter country code for this unit?

  [Unknown]:  WW

Is CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW correct?

  [no]:  yes


Enter key password for <plm-server>

        (RETURN if same as keystore password):
```

Getting server's public key certificate and storing it in client's keystore

The next step is, getting server's (self signed) public key certificate and storing it in client's keystore.

```
# keytool -export -file plm-server.crt -keystore plm-server.jks -storepass Agile123 -alias
plm-server

Certificate stored in file <plm-server.crt>

# keytool -import -file plm-server.crt -keystore plm-client.jks -storepass Agile123 -alias
plm-server

Owner: CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW

Issuer: CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW

Serial number: 50c9b2be

Valid from: Thu Dec 13 11:49:34 CET 2012 until: Wed Mar 13 11:49:34 CET 2013

Certificate fingerprints:

         MD5:  54:C8:49:EC:90:75:7D:34:FD:A2:F9:1B:2E:12:52:F0

         SHA1: F0:F7:ED:C7:14:AA:BF:BD:93:A4:7C:F0:59:7D:15:C7:94:4B:CA:B0

         Signature algorithm name: SHA1withRSA

         Version: 3

Trust this certificate? [no]:  yes

Certificate was added to keystore
```

Getting client's public key certificate and storing it in server's keystore

```
# keytool -export -file plm-client.cert -keystore plm-client.jks -storepass Agile123 -
alias plm-client

Certificate stored in file <plm-client.crt>

# keytool -import -file plm-client.cert -keystore plm-server.jks -storepass Agile123 -
alias plm-client

Owner: CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW

Issuer: CN=My Company, OU=PLM, O=My Company, L=Everywhere, ST=World Wide, C=WW

Serial number: 50c9b450

Valid from: Thu Dec 13 11:56:16 CET 2012 until: Wed Mar 13 11:56:16 CET 2013

Certificate fingerprints:

         MD5:  D6:36:E8:EB:D3:93:88:74:80:10:09:01:76:61:DE:A4

         SHA1: E6:7A:C9:BC:4A:4C:62:30:7D:07:0C:1A:83:54:65:19:1A:85:10:82

         Signature algorithm name: SHA1withRSA

         Version: 3

Trust this certificate? [no]:  yes

Certificate was added to keystore
```

Configuring the MCAD Connectors for HTTPS

To make use of the keystore client file, certain Java defines needs to be added to the Java command line (e.g. to acx.bat). This enables the Java application to validate the client certificate with the server's one and then establish an HTTPS trusted and secure connection.

The file "plm-client.jks" should be copied into the MCAD connector's "ini" directory.

Then the connector's startup script "acx.bat" needs to be modified by adding this line. The keystore password is the one that had been used with the keytool command above.

The code below shows how to use the required parameters on a Windows system only, as the CAD connectors are only supported on Windows.

```
set HTTPS_OPTS=-Djavax.net.ssl.trustStore=%CAX_ROOT%\ini\plm-client.jks -
Djavax.net.ssl.trustStorePassword=agile  -Djavax.net.ssl.keyStore=%CAX_ROOT%\ini\plm-
client.p12 -Djavax.net.ssl.keyStoreType=pkcs12 -Djavax.net.ssl.keyStorePassword=agile
```

The following line should be modified to use the HTTPS_OPTS definition. The variable "%HTTPS_OPTS%" could be included in the command line at any location, but it should be after the variable "%JAVA_HEAP_SIZE%".

```
start /b "%JAVA_HOME%\bin\javaw.exe" %JAVA_HEAP_SIZE% %HTTPS_OPTS%
-Dcom.xplm.agile.Language=EN …
```

Code sample in calling applications:

```
System.setProperty("javax.net.ssl.keyStore", "plm-client.jks");

System.setProperty("javax javax.net.ssl.keyStorePassword", "Agile123");

System.setProperty("javax.net.ssl.trustStore", "plm-client.jks");
```

This page is blank.