Agile Product Lifecycle Management

MCAD Connectors for Agile Engineering Collaboration

Administration Guide

V 3.6.2.0

Oracle Part Number – F17193-01

March 2019

Copyrights and Trademarks

This document contains information protected by copyright.

All rights are reserved, including the translation. No part of this documentation may be reproduced in any way (print, photocopy, microfilm or any other form) or processed, duplicated or distributed by use of electronic system without written permission of the company. The information contained in this documentation does not constitute any obligation on the part of the seller. The software described in this documentation is delivered under licensing contract which governs its use.

XPLM Solution is not liable for errors in this documentation.

All trademarks are the property of their respective owners.

Contact Address (for Support Services see Preface chapter):

XPLM Solution GmbH
Altmarkt Galerie Dresden, Altmarkt 25
01067 Dresden, Germany
www.xplm.com

XPLM Solution Inc.
250 Commercial Street, Suite 520
Worcester, MA 01608 USA

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.

Reserved Functionality

Note that not all parts of the MCAD connector, though accessible, are intended for customization by end-users. There is a certain set of functionality, reserved option settings and files (TCL script files, CustomerFunction.cmd) that should not be modified by end-users without consulting XPLM's services department first. This should ensure that MCAD connectors installed in productive environments always behave as expected.

Content

Installing and Upgrading Java Runtime Environments 9
Installation 9
Upgrade 9
Creo Parametric Connector Administration 10
Configuration file xAcp.cfg 10
Setup the Attribute Mapping 10
Configuration file config.pro 10
Mapping file AcpCustomer9.ini 11
Mapping Options for [ProEToAgile.XXXX] Sections 15
Mapping Options for [AgileToProE.XXXX] Sections 16
Mapping Options for [AgileGetProperties.XXX] Sections 17
Managing Format Files 19
Transfer BOM Find Number Functionality 19
File Property Holding the Agile Item Number 19
BOM Table 20
Adjusting The Column Indices 23
SOLIDWORKS Connector Administration 25
Connector Configuration Settings 25
Settings in XPlmSolidWorksConnector.xml 26
Settings in XPlmSolidWorksA9Connector.xml 29
Setup the Workspace Root 34
Setup the Java Environment and Workspace Root 34
Setup the Attribute Mapping 34
Transfer BOM Find Number Functionality 34
Solid Edge Connector Administration 36
Connector Configuration Settings 36
Settings in XPlmSolidEdgeConnector.xml 36
Settings in XPlmSolidEdgeA9Connector.xml 38
Setup the Workspace Root 42
Setup the Java Environment and Workspace Root 42
Setup the Attribute Mapping 42
EC Web Connector Administration 43
Oracle PD Cloud Functionality 43
Filtering Sub-Classes and AutoNumber Generators Displayed by the MCAD GUI 44
Preferences Settings on MCAD-CONFIG-{CAD} folder 45
Preferences Dialog 45
Load Options 46

Save Options 48
"Save as": List of automatically handled attributes 51
Class Preferences 52
Viewable Creation Preferences 53
Property Value Preferences 56
Customizing Quick View Attributes 56
CAXConfig.xml Settings 58
Basic Section 58
ConnectionProperties Section 60
BrowserDisplay Section 61
FlowControl Section 64
DateFormats Section 65
TableDisplay Section 65
WorkspaceTableDisplay Section 67
OverrideConfiguration Section 67
FileNaming Section 69
Viewables Section 70
WorkspaceDeleteViewables Section 70
PartFamilies Section 71
ChangeProperties Section 72
READONLY_FF_FIELDS Section 75
READONLY_ITEM_FIELDS Section 76
CAX_NAMES_BY_ID Section 76
Login Access Administration in PLM 76
CAD Startparts Administration in PLM 78
Creating the Template Structure in Agile 78
Adding Template Files to the Structure 80
Subtypes 83
Structure Resolution 83
Possible Errors 84
Mapping 86
Mapping Editor 86
Using the Mapping Editor 86
MCAD-MAPPING folders – How the mapping is handled 87
Mapping CAD properties to PLM fields 89
Mapping Empty Values to and from PLM Fields 93
Mapping PLM values to CAD Properties 93
Formatting values during mapping 95
Mapping Values on Save As 96
Language and Localization Administration 98

PLM User and Data Language 98
EC Web Components 98
CAD Connector Components 98
CAD Thumbnail Support 99
Transfer CAD thumbnails to PLM....100
AutoVue configuration for CAD thumbnails 100
Thumbnails in the Load Preview....100
Troubleshooting....100
Legacy Workspaces 100
Toolbox and Library Parts (ACW and ACE Only) 100
Configuring the Standard Parts Directory....101
Saving....101
Loading 101
Renaming....101
Viewables....102
Files Without Collaboration Files (.clb Files) 102
Save As....102
Strict CAD Modification Workflow (ACW and ACE only) 102
Option Settings 102
Activating the Feature 104
For SOLIDWORKS 104
For Solid Edge 104
Changes to the CAD Workflow 105
Required Server Configuration for Using MCAD Connectors 106
Activating Thumbnails 106
Designs – Page Two 107
Designs – Files....108
Designs – Structure....109
Designs – Where Used – Design 110
Designs – Relationships 111
Parts – Title Block 111
Parts – BOM....112
Parts – Relationships....114
Parts – Changes – Attributes: Pending Changes 114
Configuration when using Documents classes as Items for Drawings:....115
Documents – Relationships.... 115
Documents – Pending Changes.... 115
Documents – BOM 115
Duplicate Find Numbers 116
Allowing MCAD Engineers to See Checkout Users 116

Enabling HTTPS for Engineering Collaboration Clients 118
Introduction 118
Exporting the server certificate 118
Creating the Client Keystore for One-Way Authentication 119
Creating the Client Keystore for Two-Way Authentication 120
Configuring the MCAD Connectors for HTTPS 121
Configuring custom client applications for HTTPS 122
Enabling WSS for Engineering Collaboration Clients 122
Introduction 123
Configuring the MCAD Connectors for WSS 123
Intermediate Certificates 123
Troubleshooting 124
Lifecycle Phase Value Does not Appear for Document Objects 124

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.

Installing and Upgrading Java Runtime Environments

Installation

Since release 3.4.3.0, 3.5.0.11 and 3.6.0.0, the MCAD connector's installation package no longer includes the Java Runtime Environments (JREs) necessary for running the MCAD connector. Therefore, 32 and 64 bit JREs need to be installed prior to installing the MCAD connector on a system. However, only Java 8 is supported. The JREs in question can be downloaded using the following link: https://java.com

The Java Runtime Environments installed on the system should be detected automatically by the installation wizard of the MCAD connector. However, the user may provide the installation paths of the corresponding Java Runtime Environments manually during the installation process, if necessary or in case that the installation wizard is unable to automatically detect them.

Upgrade

The JRE installation directories are version dependent. Therefore, a JRE upgrade will result in the MCAD connector not functioning. In this case, an update of the following environment variables is required, which the MCAD connector uses to locate the JREs:

- `XPLMJavaX64`
- `XPLMJavaX86`

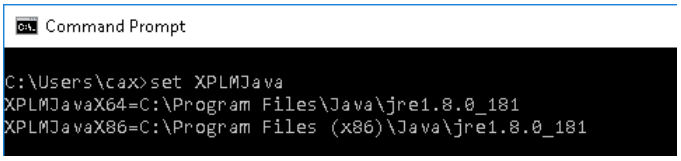


Figure 1: Environment variables used by the MCAD connector to locate the JREs

After upgrading the JREs, these two environment variables need to be updated to match the installation directory of the upgraded 32 bit and 64 bit JREs. This can be done manually or using the following script file provided within the MCAD installation directory:

```
..\XPLM Solution GmbH\OraclePLM\accessories\Set_Java_Paths.bat
```

The script file queries the system for the installation directories of the JREs and updates the environment variables mentioned above. The `Set_Java_Paths.bat` script requires administration privileges.

Creo Parametric Connector Administration

This section provides a complete summary of configuration options available for the Creo Parametric 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 Creo Parametric Connector. See the *EC Web Connector Configuration Options* section for details.

Table 1: List of all Configuration Files for the Creo Parametric Connector

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

Note Configuration files may change content between connector releases. When upgrading to a new release, incorporate the option settings from the previous release's configuration files into the new version of these files. Therefore, any text processing merging application can be used. Failure to do so might cause unintended behavior of the connector. Please contact XPLM consulting for more information on customer reserved functionality.

Configuration file xAcp.cfg

The configuration file **xAcp.cfg** contains basic system parameters.

Setup the Attribute Mapping

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

Configuration file config.pro

We recommend that you do not modify this file.

Mapping file AcpCustomer9.ini

This is the main file for controlling the behavior of the Creo Parametric 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 2: Description of all sections in xAcpCustomer9.ini

Section name	Description
Initialize	Common switches to control the behavior of the Creo Parametric 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 Creo Parametric files, during the Save command.
AgileToProE.PRT	Defines those Agile attributes that are saved automatically into Creo Parametric PRT files, during the Save command.
AgileToProE.DRW	Defines those Agile attributes that are saved automatically into Creo Parametric DRW files, during the Save command.
AgileToProE.ASM	Defines those Agile attributes that are saved automatically into Creo Parametric ASM files, during the Save command.
AgileGetProperties.PRT	Defines those Agile attributes that are saved into Creo Parametric PRT files, when using the Update Properties command.
AgileGetProperties.DRW	Defines those Agile attributes that are saved into Creo Parametric DRW files, when using the Update Properties command.
AgileGetProperties.ASM	Defines those Agile attributes that are saved into Creo Parametric ASM files, when using the Update Properties command.

The following table provides details from each section.

Table 3: [Initialize] Section Parameters

Parameter name in section [Initialize]	Parameter values	Description
AcpDebug	0 / 1 / 2 / 3 **Default: 0**	0 → no Debug 1 → write full debuginfo to ...AcpUser\log\acp.log (bad performance) 2 → write additional timer info to Creo Parametric message log (trail.txt), no acp.log is generated 3 → write only timer info to ...AcpUser\log\acp.log AcpDebug
JNI_DEBUG	0 / 1 **Default: 0**	0 → no JNI Debug 1 → write JNI debug info to ...AcpUser\log\proxy.log
AcpInitialRenameObject	1 / 0 **Default: 1**	0 → no Creo Parametric file renaming 1 → Creo Parametric file rename during initial check-in
AcpSaveDrwFrm	1 / 0 **Default: 1**	1 → Creo Parametric drawing formats (FRM) are stored in Agile 0 → ignore Creo Parametric drawing formats (FRM)
AcpSaveLay	1 / 0	1 → Creo Parametric layouts (LAY) are stored in Agile 0 → ignore Creo Parametric layouts (LAY)
AcpHelpPartIdent	ITEM **Default: ITEM**	Name of Creo Parametric 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 **Default: N**	Value that the Creo Parametric parameter should be set to in order to activate the filter

Parameter name in section [Initialize]	Parameter values	Description
AcpHelpPartSkeletonCheck	1 / 0 **Default: 1**	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 **Default: 1**	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 **Default: 1**	1 → create additional relation for components of interchange ASMs during "Save" 0 → not creating additional relation for components of interchange ASMs
AcpLoadUpdateDrwProperties	1 / 0 **Default: 1**	1 → automated call of function "Update properties" after load a DRW from Agile 0 → No action after load a DRW from Agile
AcpFamilyCheckVerify	1 / 0 **Default: 0**	0 → Do NOT check family instance verify status before saving 1 → Check family instance verify status before saving Note: Entry does not exist and has to be added manually.
AcpSaveShowMissingDrwModel	0 / 1 **Default: 0**	0 → Does NOT display external references of a simplified representation within the Save Preview 1 → Displays all objects of the simplified representation Note: Entry does not exist and has to be added manually.

Parameter name in section [Initialize]	Parameter values	Description
AcpUseCacheForId	0 / 1 **Default: 1**	0 → Does NOT check for .clb file 1 → If .clb file is deleted, the related objects are market as unknown

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 Creo Parametric. Here Creo Parametric 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 Creo Parametric, 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 Creo Parametric system attributes such as file name, object type, version of Creo Parametric that is being used, and so forth.

Table 4: Standard mapping values using "Std" prefix

Std.CreSystem	Creo Parametric version such as "Creo 3.0"
Std.VerStamp	Timestamp
Std.FileName	File name, for example "BOLT.PRT"
Std.ObjectName	Creo Parametric 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	Creo Parametric object type. Possible values are PRT, ASM, DRW, or FRM.

Par is a reference to user-defined parameter in Creo Parametric, such as `MATERIAL`, `DESCRIPTION`, or `ENGINEER`. These types of mappings are only useful where the Creo Parametric 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 5: 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, 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 is assigned to the newly created Design object.

The following example maps a Creo Parametric parameter `NAME` to the Agile attribute `DESCRIPTION` and the Creo Parametric version to Agile attribute `CAX_CRE_SYSTEM`:

Table 6: 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 Creo Parametric which occur automatically during the save process. As this adds 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 Creo Parametric 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 Creo Parametric, 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 Creo Parametric 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 Creo Parametric parameter to be updated, and the right side specifies how to find the desired row and column in the table below:

Table 7: Mapping Options with Examples

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

Section	Represents	Example
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.

Table 8: Predefined formats

Format	Description
None	no processing
ToLower	convert the value to lower case
ToUpper	convert the value to uppercase
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
```

Note Please contact XPLM consulting for more information on customer reserved functionality.

Managing Format Files

Format files (file ending .frm) provide some kind of template (including title block, borders etc.) for Creo drawing files. Usually, Creo drawings reference the format files to which they are related just as the drawing would reference any other file. The MCAD connector for Creo Parametric is capable of saving drawings including the referenced format file to Agile: Therefore, the *AcpSaveDrwFrm* option setting needs to be set to “1”. Thus, when loading from Agile, format files can be downloaded alongside a drawing file, too. However, we recommend not using Agile and the MCAD connector for managing format files, since the MCAD connector’s Create functionality does not perform any structure resolution and thus does not download format files alongside a drawing. Instead, format files should be managed using a synchronized folder, drive or network share, which is available to all CAD clients.

Transfer BOM Find Number Functionality

Since release 3.6, the MCAD connector or Creo Parametric is capable of transferring BOM find numbers to Agile. The functionality requires a certain configuration to work.

File Property Holding the Agile Item Number

In order for being able to assign find numbers to actual Item objects, each Creo CAD file needs to hold a file property (in Creo called "Parameter") containing the Item number of its corresponding Item object. Per default, the file property AGILE_PARTID is used.

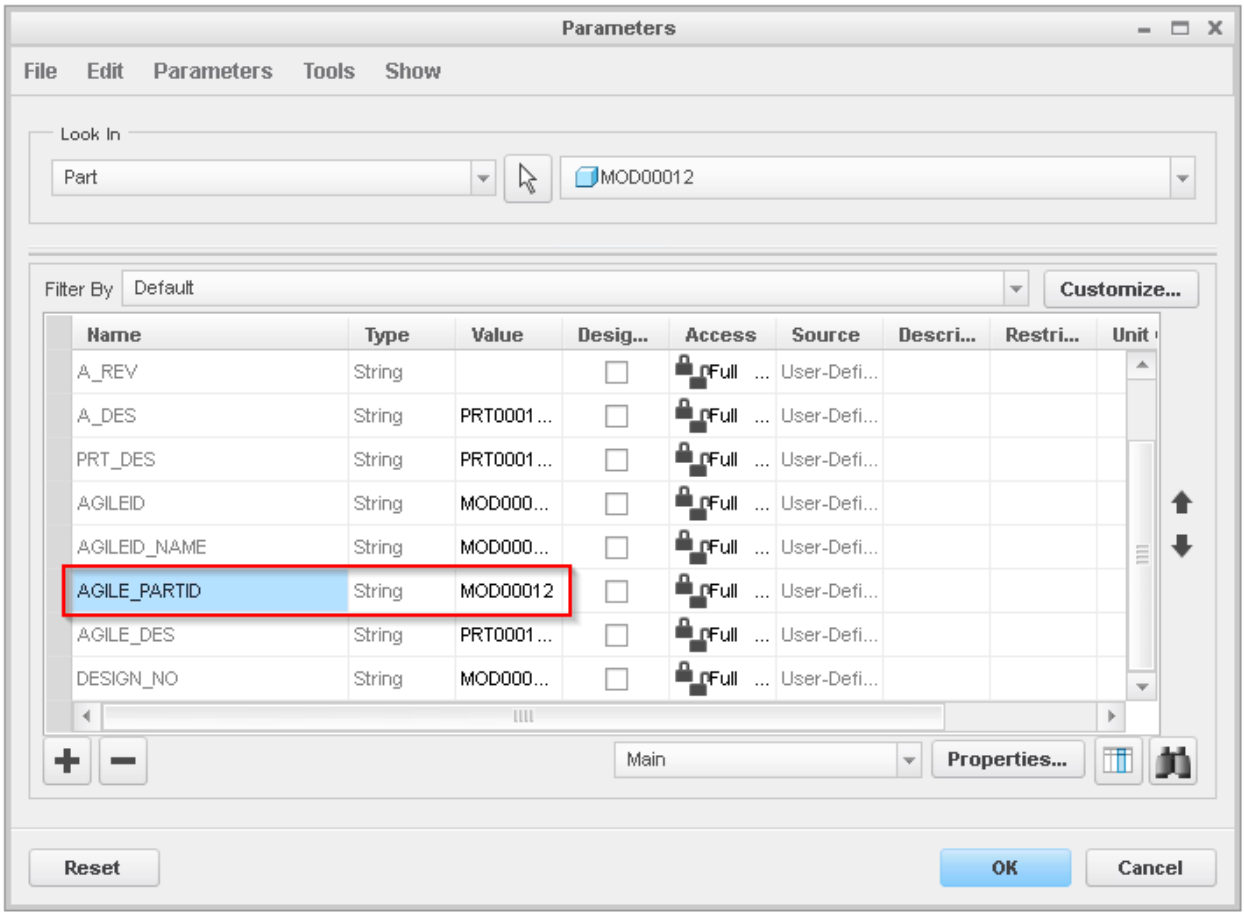


Figure 2: File property ("Parameter") holding the Item number

It is required to populate the Item number to the file property mentioned above before executing *Transfer BOM find numbers*. For the AGILE_PARTID property, it is done automatically (no mapping definition needs to be set up in the Mapping Editor) when performing *Update Properties*.

BOM Table

The next step is to create a suitable BOM table in a Creo drawing, as shown in the following screen shot. It is critical at this point that a column containing the Item number of the related Item objects exists in the table. In this case, this is the *Part ID* column.

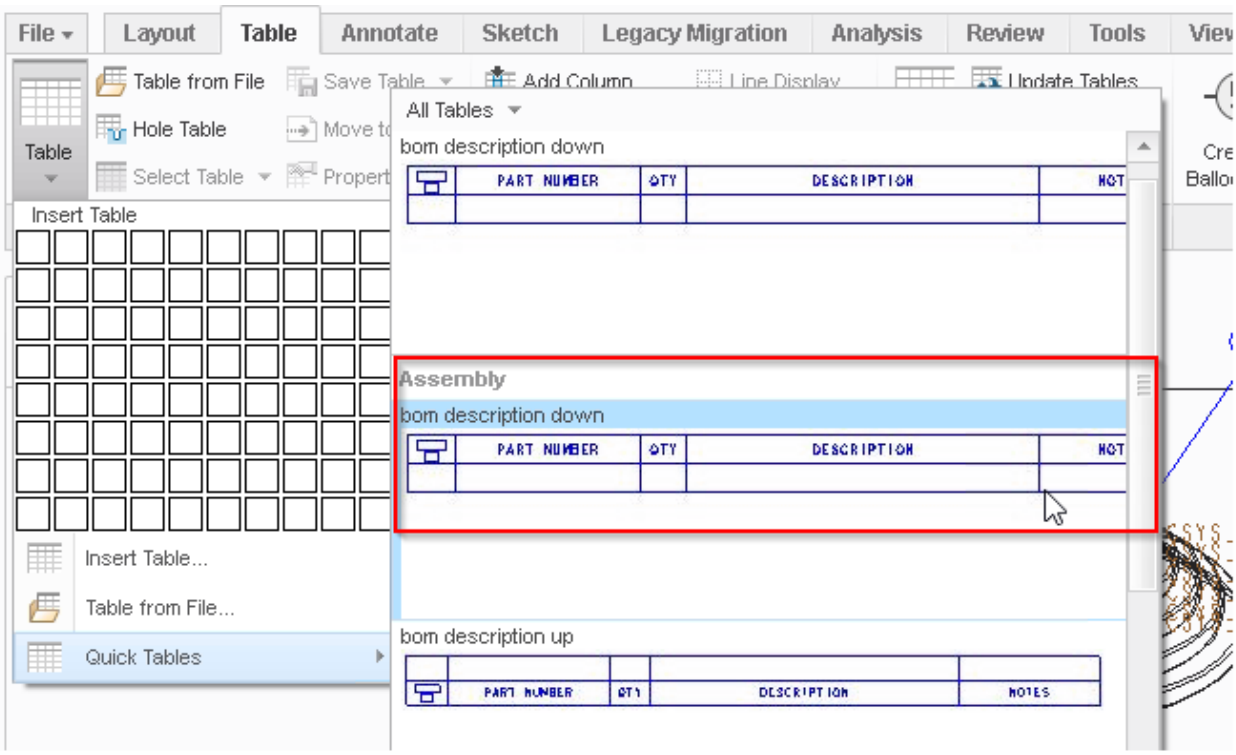
ITEM NO	Part ID	QTY	DESCRIPTION	NOTES
1	MOD00004	1	base.prt	
2	MOD00008	1	plate.prt_01	
3	MOD00009	1	plate.prt_02	
4	MOD0001D	1	plate.prt_03	
5	MOD00011	1	plate.prt_04	
6	MOD00007	1	plate.prt_05	
7	MOD00006	3	staff.prt	

Figure 3: Creo BOM table suitable for find number mapping

In addition, there must be a *find number*, *quantity* and *description* column in that table. This makes four table columns in total which are necessary for executing a find number mapping. For details on how to define which table column contains which piece of information, refer to the Adjusting The Column Indices chapter below.

To configure this column in a Creo BOM table based on a suitable Creo table template, follow the steps below:

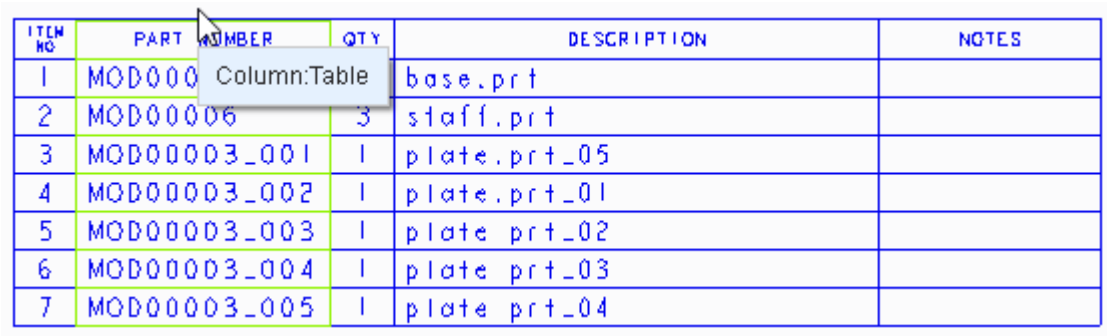
1. Table → Quick Tables → Assembly: BOM description down (double click)



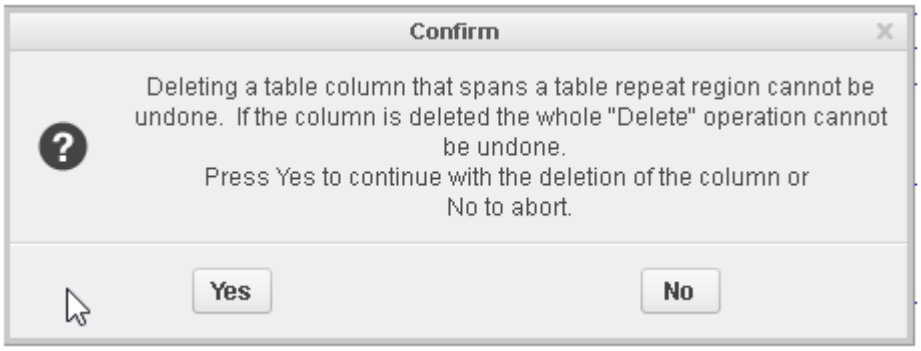
2. Place the template table in the drawing:

ITEM NO	PART NUMBER	QTY	DESCRIPTION	NOTES
1	MOD00004	1	base.prt	
2	MOD00006	3	staff.prt	
3	MOD00003_001	1	plate.prt_05	
4	MOD00003_002	1	plate.prt_01	
5	MOD00003_003	1	plate prt_02	
6	MOD00003_004	1	plate prt_03	
7	MOD00003_005	1	plate prt_04	

3. Select the existing PART NUMBER column and delete it:



Press the *delete (Del)* button. Confirm the following dialogue with "Yes":



4. Add a new column at the same position (means, same column index) where the PART NUMBER column was:

ITEM NO		QTY	DESCRIPTION	NOTES
1		1	base.prt	
2		1	plate.prt_01	
3		1	plate.prt_02	
4		1	plate.prt_03	
5		1	plate.prt_04	
6		1	plate.prt_05	
7		3	staff.prt	

5. Add the necessary file property reference to the newly created table column. Therefore, edit the first cell in that table (select the cell, use *Table* → *Properties*) and add the text value **`&asm.mbr.AGILE_PARTID`** to it.

6. Press *Table* → *Update Tables*. If everything works as expected, the Agile Item numbers should now be visible in the table column (also adjust the table header and column width as necessary):

ITEM NO	Item Number	QTY	DESCRIPTION	NOTES
1	MOD00004	1	base.prt	
2	MOD00008	1	plate.prt_01	
3	MOD00009	1	plate.prt_02	
4	MOD00010	1	plate.prt_03	
5	MOD00011	1	plate.prt_04	
6	MOD00007	1	plate.prt_05	
7	MOD00006	3	staff.prt	

Adjusting The Column Indices

With the example table given above, find number mapping from Creo to Agile should work. However, it is possible to adjust the MCAD connector for working with different table layouts. Therefore, it is possible to adjust the column indices for each of the columns required by the MCAD connector on a BOM table which is supported for find number mapping.

AcpCustomer9.ini, [AcpDrwBomTab.DEFAULT] section:

```
# Column mapping
AcpDrwBomTabColumnNUMBER          =     2
AcpDrwBomTabColumnFIND_NO         =     1
AcpDrwBomTabColumnQUANTITY        =     4
AcpDrwBomTabColumnDESCRIPTION     =     3
```

Table 9: Option settings for find number mapping in AcpCustomer9.ini

Setting	Purpose and available values
AcpDrwBomTabFirstRow	Row index of the first table row after the table header. This is usually the second row. **Default**: **2** **Possible values: Any positive integer value**
AcpDrwBomTabColumnNUMBER	Column index for the Item number column**.** **Default**: **2** **Possible values: Any positive integer value**
AcpDrwBomTabColumnFIND_NO	Column index for the find number column. **Default**: **1** **Possible values: Any positive integer value**
AcpDrwBomTabColumnQUANTITY	Column index for the quantity number column. **Default**: **4** **Possible values: Any positive integer value**
AcpDrwBomTabColumnDESCRIPTION	Column index for the description column. **Default**: **3** **Possible values: Any positive integer value**

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

XPlmSolidWorksA9Connector.xml - contains mostly Agile PLM related settings

XPlmSWAgileAdd-in.xml – contains the Agile menu definition and Add-in 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**.

Settings in XPlmSolidWorksConnector.xml

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

Note: Options which are not supported yet are not listed in the following table.

Table 10: Settings

Setting	Purpose and available values
EnableSolidWorksLogging	If set to true, logging is enabled. **Default**: **false** **Possible values: true | false**
SolidWorksLogFile	Value is the full path to the log file, required if logging is enabled. **For example** C:\SolidWorks.log
SolidWorksLogLevel	Defines the level of detail of the log file. A higher number indicates more log messages. **Default: 10** **Possible values: 10 | 100**
SolidWorksCreateUniqueFileNames	If set to true, unique file names are created. Value false has no special logic. **Default**: **true** **Possible values: true | false**
SolidWorksUseLocalFileCache	If set to true, the local cache is used. **Default**: **true** **Possible values: true | false**
AllowRecursiveStructure	If set to true, transfer of recursive structure is allowed. **Default**: **true** **Possible values: true | false**
SolidWorksDefaultSaveName_1 SolidWorksDefaultSaveName_2 SolidWorksDefaultSaveName_3	If the CAD file isn't saved to hard disk at all and the integration does the first local save to disk, the integration uses a default save name defined in configuration parameter SolidWorksDefaultSaveName If the default save name still exists in the working folder, the integration adds automatically an increment number. **Default SolidWorksDefaultSaveName_1: Part.sldprt** **Default SolidWorksDefaultSaveName_2: Assembly.sldasm**

Setting	Purpose and available values
	Default SolidWorksDefaultSaveName_3: Drawing.slddrw
SWAddins	Additional Add-ins to be loaded. **Default**: **empty**
SWAddinToolbarTitle	Name of the XPLM Add-in **Default: XPLM Solution**
IgnoreMissingParts	If set to true, missing parts are ignored. If set to false, an error message is thrown for missing parts. **Default**: **true** **Possible values: true | false**
SolidWorksCSVSeparator	It is possible to import CAD object meta data via CSV data. CSV file needs same file name like CAD object. Therefor this property defines the separator in the CSV file. **Default**: **@** **Possible values: any special character**
SolidWorksCreateBitmapPreview	If set to true, preview bitmaps are created on save. **Default**: **false** **Possible values: true | false**
SolidWorksCreateBitmapPNGPreview	Controls whether (true) or not (false) PNG preview images are created on save. **Default**: **false** **Possible values: true | false**
SolidWorksSearchOptionPath	Allows the manipulation of the SOLIDWORKS *search path* (*Options → File Paths → Search Paths*) during integration load processes. The path given in this option setting is added to the standard search path. Any valid directory path can be used as option value. **Default: -no value-**
SolidWorksCommandTabName	String to display. **Default**: **xPLM Solution**
SolidWorksScriptEngine	Do not change! **Default**: **intern**
SolidWorksEvent_StartNotify	If set to true, integration starts up immediately to preserve memory. If set to false, the integrations starts on demand. **Default**: **true**

Setting	Purpose and available values
	Possible values: true | false
SolidWorksEvent_PartModifyNotify	If set to true, user will be informed when modifying a part which is not checked out. **Default: false** **Possible values: true | false**
SolidWorksEvent_AssemblyModifyNotify	If set to true, user will be informed when modifying an assembly which is not checked out. **Default: false** **Possible values: true | false**
SolidWorksEvent_DrawingModifyNotify	If set to true, user will be informed when modifying a drawing which is not checked out. **Default: false** **Possible values: true | false**
SolidWorksTiffDPI	Dots per inch of tiff **Default: 300**
SolidWorksStandardPartDir	Defines the location of toolbox (standard) parts. Refer to the *Toolbox and Library Parts (ACW and* ACE Only) chapter for details. **Default: C:\SolidWorks Data**

Settings in XPlmSolidWorksA9Connector.xml

This section describes the available settings and valid values for the connector in **XPlmSolidWorksA9Connector.xml.**

Note: Options which are not supported yet are not listed in the following table.

Table 11: Settings

Setting	Purpose and available values
EnableScriptEngineLogging	If set to true, logging is enabled. **Default**: **false** **Possible values: true | false**
ScriptengineLogFile	Value is the full path to a log file, required if logging is enabled. If you do not specify a path the log file is written to the user home AgileCache folder, for example C:\caxlog\xacw.log. **Default: xacw.log**
ScriptengineLogLevel	Defines the level of detail of the log file. A higher number indicates more log messages. **Default: 10** **Possible values: 10 | 100**
SolidWorksMenuFiles	The add-in menu file in the xml directory **Default**: **XPlmSWAgileAdd-in.xml**
RenameOnLoad	NUMBER: Rename file names are equal to the PLM number. CAX_FIL_NAME: Do not rename on load. **Default: CAX_FIL_NAME** **Possible Values: Any other value**
RebuildOnLoad	If set to true, a rebuild is executed after downloading CAD files. **Default: false** **Possible values: true | false**
RenameOnInitialSave	If set to true, files are renamed on initial save. **Default**: **false/** **Possible values: true | false**
RenameOnSave	If set to true, files are renamed on save as. **Default**: **false/** **Possible values: true | false**

Setting	Purpose and available values
RenamingRule	Renaming rule for building the filename. This is a reserved option setting, always set the default value. **Default %CAX_NEW_NAME%**
SolidWorks_AllowReopenOnRename	If set to true, renamed files are reopened in CAD. **Default: true** **Possible values: true | false**
SolidWorks_RemovePLMUnknownDrawings AfterRenameOnSave	If set to 1, PLM unknown drawings are removed from workspace after rename on initial save. If set to 0, no drawings are removed from workspace after rename. **Default: 0** **Possible values: 0 | 1**
ConfiguredDefault	Controls the default behavior 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** **Possible values: true | false**
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**: **Configured2**
ConfiguredValue_Configured	Optional value of the ConfiguredProperty or ConfiguredProperty2 that would be interpreted as “Yes” (configured). **Default: yes** **Possible values: yes | no**
ConfiguredValue_NotConfigured	Optional value of the ConfiguredProperty or ConfiguredProperty2 that would be interpreted as “No” (not configured). **Default: no** **Possible values: yes | no**
MasterConfigProperty	If the given property name is contained in the configuration specific

Setting	Purpose and available values
	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**
FindPLMObject	Whether to search for existing configurations and files in PLM. Allowed values are true and false. **Default: true** **Possible values: true | false**
SolidWorks_DisableUpdateDrawingBOM	Whether to reset the update flag in drawings to suppress updating the parts list on Load. Allowed values are true and false. **Default: false** **Possible values: true | false**
SolidWorks_AlwaysExtractExternalReferences	If set to true, are always traversed. If set to false, external references are traversed on demand with user prompt only. **Default: false** **Possible values: true | false**
AppendPLMFieldsToViewableNames	Whether to append additional PLM fields into the viewables filenames. Allowed values are true and false. **Default: true** **Possible values: true | false**
AppendingRuleViewables	FieldID or CAX fieldnames to append. Format like **Default: %REVISION%**
SolidWorks_BulkLoad_AllConfigurations	Bulkloader switch to create all or only use configurations on initial import. Allowed values are true and false. **Default: false** **Possible values: true | false**
SolidWorks_BulkLoad_CollectDrawings	If set to true, drawings of related model are collected during bulk load to be saved to PLM, too. If set to false, integration does not search for related drawings during bulk load. **Default: true** **Possible values: true | false**
SolidWorks_InstanceViewables	Viewable type to create for configurations that have no real 3D model

Setting	Purpose and available values
	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 behavior. Allowed values are X_T and false. **Default: false** **Possible values: true | false**
Solidworks_RebuildDrawingOnSave	Perform a rebuild on saving a drawing. Should be set to true if sometimes model attributes do not reflect immediately into drawing title blocks. **Default: false** **Possible values: true | false**
SolidWorks_AlwaysSaveTopAssembly	If set to true, the top assembly will always be saved. **Default: false** **Possible values: true | false**
SolidWorks_ReserveBeforeModify_EnableUseCase	0: Use case "modify only CAD files which were checked out" is disabled, thus, status of CAD files (read only/editable) will not be changed by Agile EC .-> Load option "Read Only" and the following options from **XPlmSolidWorksA9Connector.xml** will be ignored: SolidWorks_ReserveBeforeModify_NotifyUser, SolidWorks_ReserveBeforeModify_AutoReserve 1: Use case "modify only CAD files which were checked out" is enabled. -> Load option "Read Only" and the following options from **XPlmSolidWorksA9Connector.xml** will be evaluated and work like documented: SolidWorks_ReserveBeforeModify_NotifyUser, SolidWorks_ReserveBeforeModify_AutoReserve. During check out/cancel check out the file status will be adapted (read only/editable) depending from resulting checkout status. **Default: 0** **Possible values: 0 | 1**
SolidWorks_ReserveBeforeModify_NotifyUser	If it is set to 1 and SolidWorks_ReserveBeforeModify_EnableUseCase is also set to 1, the user will be informed when modifying files which are not checked out.

Setting	Purpose and available values
	Default: 0 **Possible values: 0 | 1** **Default: 0**
SolidWorks_ReserveBeforeModify_AutoReserve	If it is set to 1 and SolidWorks_ReserveBeforeModify_EnableUseCase is also set to 1, files which are modified will be reserved automatically. **Default: 0** **Possible values: 0 | 1**

Setup the Workspace Root

The Java environment and workspace root are set in **<install dir>\com\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 **<install dir>\com\start_acw.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_acw.bat** during the initial registration and also on startup of the connector.

Setup the Attribute Mapping

Use the *Mapping Editor* to define the attribute mapping.

Transfer BOM Find Number Functionality

In order to execute a find number mapping in the ACW integration, the **createBOM_Entry** transaction needs to be configured in *xPLMAgile9SolidWorksTransaction.xml* to match the BOM table layout used for the drawing BOM tables.

The following field definitions need to be given in that transaction:

Field Name	Required Configuration for the *Attribut* [sic!]
FIND_NO	String given in the column header of the BOM table column containing the find numbers.
QUANTITY	String given in the column header of the BOM table column containing the quantities.
DESCRIPTION	String given in the column header of the BOM table column containing the item descriptions.

Any number of field definitions for the same column may be added to the transaction, as shown in the example below.

Example configuration:

```
<Transaction>
  <Aliasname>createBOM_Entry</Aliasname>
  <Import>
    <Parameter>
      <FieldCollection>
        <Field>
          <Name>FIND_NO</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>ITEM NO.</Attribut>
        </Field>
        <Field>
          <Name>FIND_NO</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>POS-NR.</Attribut>
        </Field>
        <Field>
          <Name>NUMBER</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>PART NUMBER</Attribut>
        </Field>
        <Field>
          <Name>NUMBER</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>BENENNUNG</Attribut>
        </Field>
        <Field>
          <Name>QUANTITY</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>QTY.</Attribut>
        </Field>
        <Field>
          <Name>QUANTITY</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>MENGE</Attribut>
        </Field>
        <Field>
          <Name>1002</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>DESCRIPTION</Attribut>
        </Field>
        <Field>
          <Name>DESCRIPTION</Name>
          <Type>ParameterList</Type>
          <Subtype>FieldCollection</Subtype>
          <Attribut>DESCRIPTION</Attribut>
        </Field>
```

```
      </FieldCollection>
      <StructureCollection>
      </StructureCollection>
      <TableCollection>
      </TableCollection>
    </Parameter>
  </Import>
</Transaction>
```

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 Solid Edge 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

XPlmSolidEdgeA9Connector.xml - contains mostly Agile PLM related settings

XPlmSEA9Add-in.xml – contains the Agile menu definition and Add-in 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.**

Settings in XPlmSolidEdgeConnector.xml

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

Note: Options which are not supported yet are not listed in the following table.

Table 12: Settings

Setting	Purpose and available values
EnableSolidEdgeLogging	If set to true, logging is enabled.

Setting	Purpose and available values
	Default: **false** **Possible values: true | false**
SolidEdgeLogFile	Value is the full path to a log file, required if logging is enabled for example C:\caxlog\SolidEdge.log
SolidEdgeLogLevel	Defines the level of detail of the log file. A higher number indicates more log messages. **Default: 10** **Possible values: 10 | 100**
SolidEdgeIgnoreMissingParts	If set to true, missing parts are ignored. If set to false, an error message is thrown for missing parts. **Default**: **true** **Possible values: true | false**
SolidEdgeDefaultSaveName_1 SolidEdgeDefaultSaveName_2 SolidEdgeDefaultSaveName_3 SolidEdgeDefaultSaveName_4	If the CAD file isn't saved to hard disk at all and the integration does the first local save to disk, the integration uses a default save name defined in configuration parameter SolidWorksDefaultSaveName If the default save name still exists in the workspace folder, the integration adds automatically an increment number. **Default SolidWorksDefaultSaveName_1: Part.par** **Default SolidWorksDefaultSaveName_2: Draft.dft** **Default SolidWorksDefaultSaveName_3: Assembly.asm** **Default SolidWorksDefaultSaveName_4: Sheet Metal.psm**
SolidEdgeCreateBitmapPreview	If set to true, preview bitmaps are created on save. **Default**: **false** **Possible values: true | false**
SolidEdgeCreateUniqueFileNames	If set to true, unique file names are created. Value false has no special logic. **Default**: **true** **Possible values: true | false**
SolidEdgeEvent_StartNotify	If set to true, integration starts up immediately to preserve memory. If set to false, the integrations starts on demand. **Default**: **true** **Possible values: true | false**
SolidEdgeEvent_AfterActiveDocumentChange	User gets information when modifying document which is not

Setting	Purpose and available values
	checked out by the user. Message appears on changing active document. **Default: false** **Possible values: true | false**
SolidEdgeStandardPartDir	Option setting is reserved for future use.

Settings in XPlmSolidEdgeA9Connector.xml

This section describes the available settings and valid values for the connector in **XPlmSolidEdgeA9Connector.xml**.

Note: Options which are not supported yet are not listed in the following table.

Table 13: Settings

Setting	Purpose and available values
SolidEdge_EnableScriptEngineLogging	If set to true, logging is enabled. **Default**: **false** **Possible values: true | false**
SolidEdge_ScriptengineLogFile	Value is the full path to a log file, required if logging is enabled. If you do not specify a path the log file is written to the user home AgileCache folder, for example C:\caxlog\xace.log. **Default: xace.log**
SolidEdge_ScriptengineLogLevel	Defines the level of detail of the log file. A higher number indicates more log messages. **Default: 10** **Possible values: 10 | 100**
SolidEdgeMenuFiles	The add-in menu file in the xml directory **Default**: **XPlmSEA9Addin.xml**
SolidEdge_SuppressUnusedMembers	If set to true, unused members are suppressed on normal save. If set to false, all available members are shown on normal save. **Default**: **true** **Possible values: true | false**
SolidEdge_SuppressTemplateMemberLinks	If set to true, circular reference are suppressed to all members.

Setting	Purpose and available values
	If set to false, external references are shown to all members. Needs SuppressUnusedMembers to be set to false as well **Default**: **true** **Possible values: true | false**
SolidEdge_RenameOnLoad	NUMBER: Rename file names are equal to the PLM number. CAX_FIL_NAME: Do not rename on load. **Default: CAX_FIL_NAME** **Possible Values: Any other value**
SolidEdge_RenameOnInitialSave	If set to true, files are renamed on initial save. **Default**: **false/** **Possible values: true | false**
SolidEdge_RenameOnSave	If set to true, files are renamed on save as. **Default**: **false/** **Possible values: true | false**
SolidEdge_RenamingRule	Renaming rule for building the filename. This is a reserved option setting, always set the default value. **Default %CAX_NEW_NAME%**
SolidEdge_RemovePLMUnknownDrawingsAfterRenameOnSave	If set to 1, PLM unknown drawings are removed from workspace after rename on initial save. If set to 0, no drawings are removed from workspace after rename. **Default: 0** **Possible values: 0 | 1**
SolidEdge_RenameUpperCase	Is used to capitalize the file names on renaming. This option actually works if checked in the workspace. Solid Edge's Feature Tree, however, does not display any assembly component in upper case except for the root element. **Default: true** **Possible values: true | false**
SolidEdge_AutomaticAutonumberOnSaveAs	If set to true, Save As Override dialog automatically retrieves an AutoNumber. If set to false, the Save As Override dialog does not automatically retrieve an AutoNumber.

Setting	Purpose and available values
	Default: false **Possible values: true | false**
SolidEdge_BulkLoad_ExcludeTraverseExternalReference	If set to true, external references are not traversed during bulk load. **Default: true** **Possible values: true | false**
SolidEdge_UseOccurrenceIncludeInBom	If set to true, user quantity from Solid Edge occurrences is used. **Default**: **true** **Possible values: true | false**
SolidEdge_PreventLastWriteTimeInfoOnOpen	If true, the file info of each downloaded file will be manipulated after CAD open to reset the 'last write time' to the value before the CAD open action. **Default: true** **Possible values: true | false**
SolidEdge_ReserveBeforeModify_EnableUseCase	0: Use case "modify only CAD files which were checked out" is disabled, thus, status of CAD files (read only/editable) will not be changed by Agile EC .-> Load option "Read Only" and the following options from **XPlmSolidEdgeA9Connector.xml** will be ignored: SolidEdge_ReserveBeforeModify_NotifyUser, SolidEdge_ReserveBeforeModify_AutoReserve 1: Use case "modify only CAD files which were checked out" is enabled. -> Load option "Read Only" and the following options from **XPlmSolidEdgeA9Connector.xml** will be evaluated and work like documented: SolidEdge_ReserveBeforeModify_NotifyUser, SolidEdge_ReserveBeforeModify_AutoReserve. During check out/cancel check out the file status will be adapted (read only/editable) depending from resulting checkout status. **Default: 0** **Possible values: 0 | 1**
SolidEdge_ReserveBeforeModify_NotifyUser	If it is set to 1 and SolidEdge_ReserveBeforeModify_EnableUseCase is also set to 1, the user will be informed when modifying files which

Setting	Purpose and available values
	are not checked out. **Default: 0** **Possible values: 0 | 1**
SolidEdge_ReserveBeforeModify_AutoReserve	If it is set to 1 and SolidEdge_ReserveBeforeModify_EnableUseCase is also set to 1, files which are modified will be reserved automatically. **Default: 0** **Possible values: 0 | 1**

Setup the Workspace Root

The Java environment and workspace root are set in **<install dir>\com\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 **<install dir>\com\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

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.

Table 14: Mapping Setup: Available Attributes

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 Solid Edge 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 log file, the available names are separated 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.

Oracle PD Cloud Functionality

Since release 3.6.1, the MCAD connector comes with a special mode of operation for Oracle PD Cloud. The option setting is available for MCAD administrators only in the *PLM Mode* drop-down list in the Preferences dialog.

Table 15: Option settings for the PLM Mode drop down list explained

Oracle PLM	Normal connector behavior for Oracle Agile PLM. This is the default functionality for Oracle Agile PLM known from previous releases of the MCAD connector. Item and Item Change creation works normally.
CAD4Cloud	Alternative connector behavior when using Oracle PD Cloud. This mode of operation hides most Item related controls in the MCAD connector GUI and is intended to be used when using the MCAD connector to connect to PD Cloud via an Agile server. When activated, this mode prevents the MCAD connector from creating and processing Items and Item Changes. Item related information are not displayed and also not transferred to or from Agile.
Oracle PLM Hybrid Cloud	Works like Oracle PLM mode, but uses PD Cloud hyperlinks for the Item related controls of the MCAD connector's GUI instead of Agile PLM hyperlinks.

The Fusion URL text field is used to define the URL that points to the PD Cloud service. The MCAD connector uses the URL given in this text field to navigate to Item objects stored in PD Cloud in *CAD4Cloud* and *Hybrid Cloud* modes.

Filtering Sub-Classes and AutoNumber Generators Displayed by the MCAD GUI

The MCAD connector allows administrators to filter out Design, Item and Change sub-classes as well as AutoNumber generators which should not be displayed in the GUI. If a filter setting is activated, any MCAD connector control displaying Design, Item or Change sub-classes or AutoNumber generators only displays the objects given in the corresponding filter definition. This way, PLM object sub-classes and AutoNumbers can be hidden from MCAD connector users to prevent unintended usage. Additionally, filtering can be file type (file ending) specific. For any CAD file ending a separate set of filter criteria can be defined, if required.

The definition of the filter criteria is performed in the **CaxConfig.xml** configuration file. Refer to chapter *OverrideConfiguration* Section on page 67 for details.

Preferences Settings on MCAD-CONFIG-{CAD} folder

The preferences are stored in PLM in a Design object called MCAD-CONFIG followed by the CAD suffix (for example MCAD-CONFIG-PROE for Creo Parametric). The user needs the Administrator role in PLM assigned in order to update the template **Preferences.xml**.

The **Preferences.xml** is stored locally first and only if you are an administrator the template is 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-{CAD} Design object in Web Client, adding your local **Preferences.xml** to the files tab, and checking the object back in after upload. The next time a user logs in, the new template is downloaded. In case the system cannot generate the MCAD-CONFIG-{CAD} Design object automatically, create it with this name in the PLM system.

To lock an entry from user modification, you need to edit the **Preferences.xml**. The template in PLM must also be replaced manually. Search for a section called “GeneralDefaults”, which contains several *FieldCollections*. Each collection describes one default. There are 3 fields with name/value pairs for each default:

Table 16: Preferences Settings

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 editable=false, even the administrator gets a setting, that is not editable anymore.

Preferences Dialog

The preferences dialog is accessed using the Preferences menu item in the CAD main menu.

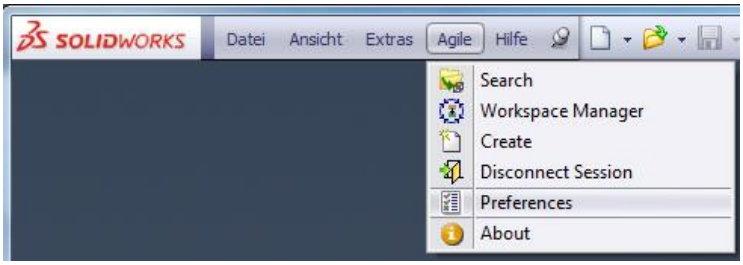


Figure 4: CAD Menu: Agile: Preferences

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

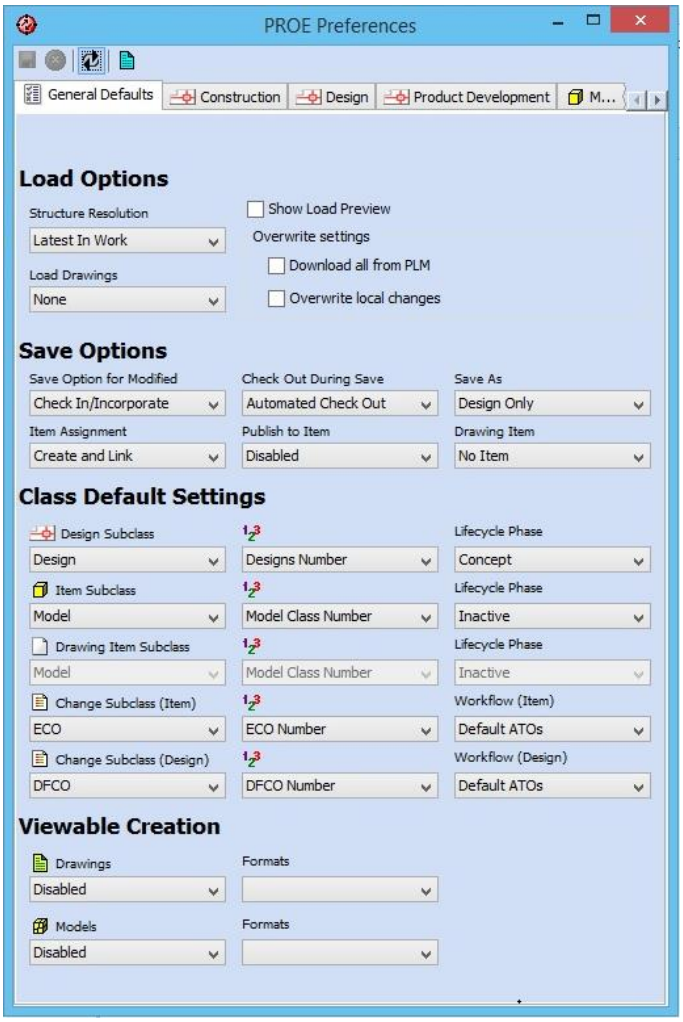


Figure 5: CAD Preferences

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 Options

The default structure resolution on load is configured using the *Structure Resolution* option setting. This defines which versions of children in Design structures are used when loading a Design structure from

Agile.

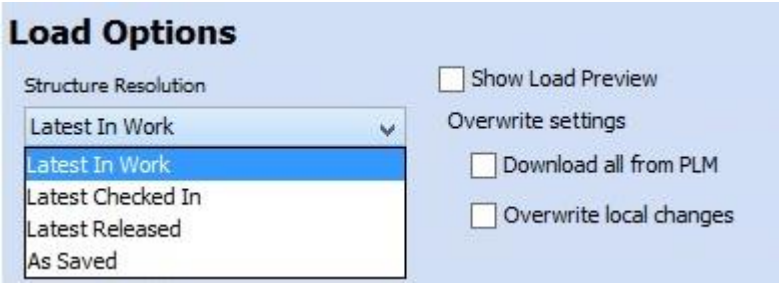


Figure 6: Load Options

The valid values and their meanings are described in the following table.

Table 17: Process Options: Description

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.

In addition, the Load Options section provides a number of load related option settings. Refer the following table.

Show Load Preview	Activates and de-activates the Load Preview window.
Download all from PLM	Orders the MCAD connector to download all files related to a Design structure when loading, even if some of the files might already be available in the local workspace directory.
Overwrite local changes	Allows the MCAD connector to overwrite locally modified files when loading from Agile. A confirmation dialog is displayed, if MCAD intends to overwrite a file.

Save Options

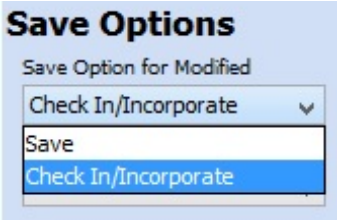


Figure 7: Default Save Option

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

Table 18: Default Save Options: Description

Save	The file is saved to the currently checked out Design version. After that, the Design is checked in and checked out again immediately. This way the Design remains checked out after save with an incremented version.
Check In/Incorporate	The file is saved to the currently checked out Design version. After that, the Design is checked in. If a pending Design Change Order is assigned to the Design object, the *Incorporate* flag is implicitly set (Design is incorporated).

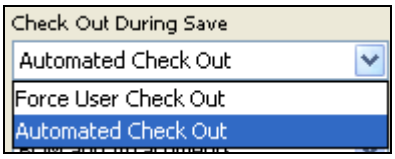


Figure 8: Checkout during Save

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

Table 19: Checkout during Save: Description

Automated Check Out	The design is 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.

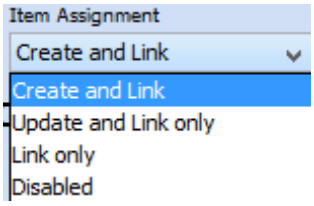


Figure 9: Part Assignment

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

Table 20: Part Assignment Options: Description

Create and Link	This creates new Part objects, if a new Design is created. The Part is linked to the Design and the Part properties are updated.
Update and Link only	This does not create Parts, but existing Parts are linked to the Design and the Part properties are updated.
Link only	This does not create or update Parts. Only the relationship link between the part and the design is created.
Disabled	Part assignment or creation is disabled completely.

Figure 10: Save As Behavior

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

Table 21: Save As Behavior Options: Description

Design Only	No Part objects are created.
Part and Design	AutoNumber is 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*.

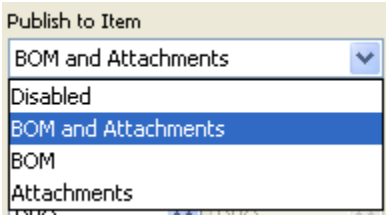


Figure 11: Publish Behavior

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:

Table 22: Publish Behavior Options: Description

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

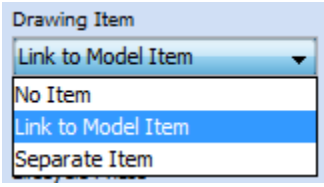


Figure 12: Drawing Item Behavior

The *Drawing Item Option* controls whether or not an Item is created for a Drawing. The valid options and their meanings are:

Table 23: Drawing Item Behavior Options: Description

No Item	No relationship is created between the Drawing Design and the Model Item.
Link to Model Item	An additional relationship between the Drawing Design and the Model Item is created.
Seperate Item	An additional Item is created for Drawing Designs. The Drawing Item Subclass is defined in the Drawing Item Subclass section.

“Save as”: List of automatically handled attributes

The following attributes are handled automatically and should not be mapped.

Titleblock tab page 1, page 2:

- Design number
- Revision
- Version
- Check out status
- CAD filename
- Design system
- File type
- Part number

Relationship tab:

- Link type

Structure tab:

- Identifier
- Component
- Revision
- Version
- Quantity

Class Preferences

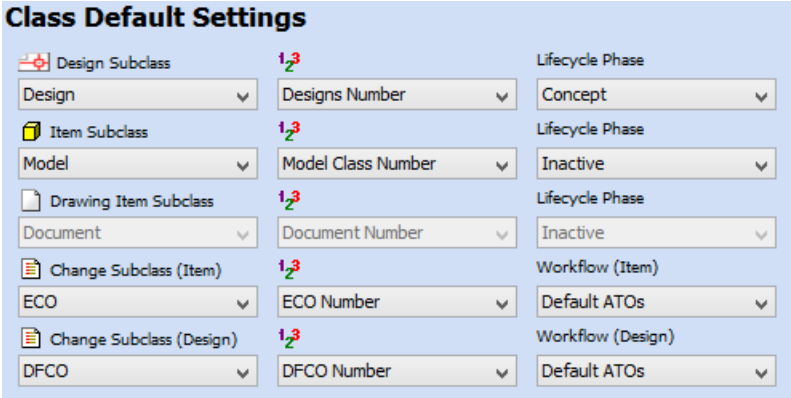


Figure 13: Class Preferences

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.

The Drawing Item Subclass settings become active only if the “Separate Item” option is enabled in the Drawing Item option.

Viewable Creation Preferences

Viewable file creation preferences determine the types of viewable files that are 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;PDF</Value></Field>
      </FieldCollection>
</Structure>
…
```

Basically, 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 on the bottom.

Figure 14: Viewable Creation

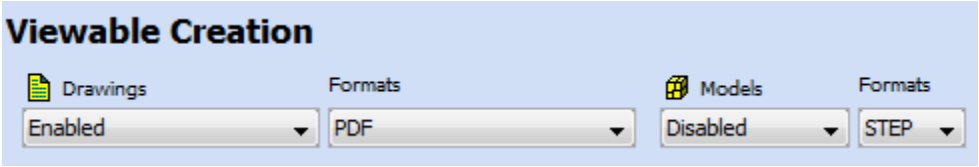


Table 24: Viewable Creation: Available Options

CAD Tool	Drawing formats	Model formats
SOLIDWORKS	PDF	STEP X_T IGES

Solid Edge	PDF	STEP JT
Creo Parametric	PDF	STEP IGES X_T (needs license for converter scripts) PDF (3DPDF, due to limitations only created for the root element of the assembly structure currently open in Creo, which is equal to the *Top CAD File* option mentioned below)

MCAD connectors use the CAD tool's PDF generation by default. So any settings to control what ends up in the PDF should be described in the CAD tool's documentation.

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 which situations the viewables selected in the *Formats* combo boxes should be created.

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

Table 25: Drawings Combo box Options: Description

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

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

Table 26: Model Combo box Options: Description

Option	Explanation
All Assemblies	If viewable formats are selected, viewables are created only for assembly

	files. **Note that 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	If viewable formats are selected, viewables are created only for part files. **Note that 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	If viewable formats are selected, viewables are created for all CAD files. **Note that 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	If viewable formats are selected, viewables are created only for the root element of a CAD file structure (as seen in the Tree View). **Note that 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.

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 set up the mapping interactively. The preferences are 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.

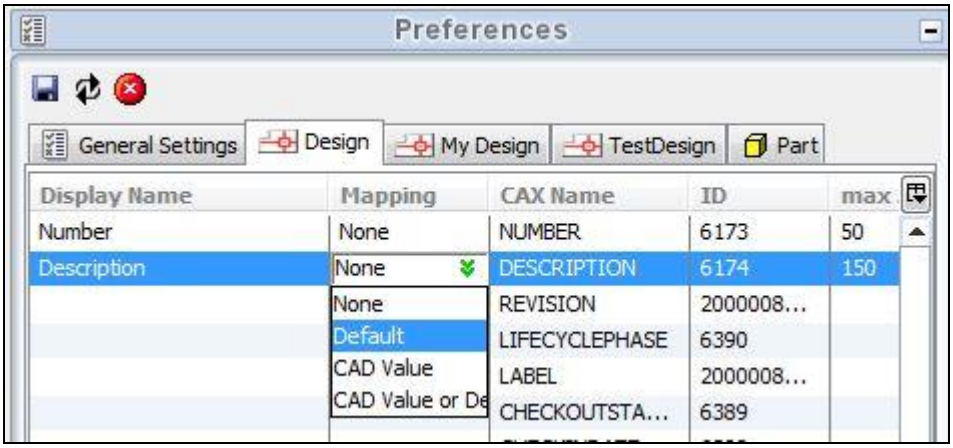


Figure 15: Property Value Preferences

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*.

Customizing Quick View Attributes

The MCAD connector supports customizing the attributes, which are displayed in the Quick View dialog window. In order to change the appearance, the following XML structure needs to be added to CaxConfig.xml:

```
<Structure>
      <Name>QuickViewDisplay</Name>
      <FieldCollection>
            <Field><Name>CAX_FIL_NAME</Name><Value>1</Value></Field>
            <Field><Name>COMPONENTTYPE</Name><Value>1</Value></Field>
            <Field><Name>NUMBER</Name><Value>1</Value></Field>
            <Field><Name>REV</Name><Value>1</Value></Field>
            <Field><Name>REVISION</Name><Value>1</Value></Field>
            <Field><Name>DESCRIPTION</Name><Value>1</Value></Field>
            <Field><Name>LABEL</Name><Value>1</Value></Field>
            <Field><Name>LIFECYCLEPHASE</Name><Value>1</Value></Field>
            <Field><Name>CHECKOUTUSER</Name><Value>1</Value></Field>
            <Field><Name>Item.NUMBER</Name><Value>1</Value></Field>
            <Field><Name>Item.REV</Name><Value>1</Value></Field>
            <Field><Name>Item.ECO</Name><Value>1</Value></Field>
            <Field><Name>Item.LIFECYCLEPHASE</Name><Value>1</Value></Field>
            <Field><Name>Item.DESCRIPTION</Name><Value>1</Value></Field>
            <Field><Name>CAX_MODEL_TYPE</Name><Value>1</Value></Field>
            <Field><Name>CAX_MODEL_REF</Name><Value>1</Value></Field>
            <Field><Name>CAX_LINK_TYPE</Name><Value>1</Value></Field>
            <Field><Name>CAX_LINK_REF</Name><Value>1</Value></Field>
            <Field><Name>CAX_TYPE</Name><Value>1</Value></Field>
      </FieldCollection>
</Structure>
```

Additional attributes can be added to the structure as required. Make sure to assign the value 1 to each XML *Field*. Note, that the Quick View dialog does only display attributes, which are set to visible in the class configuration of the Preferences dialog.

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>
```

Table 27: Basic CAXConfig.xml Settings: Description

Setting	Purpose and available values
SITE	Defines the global parameter for the MCAD connector configuration. MCAD connectors with different SITE setting use different configuration objects in Agile. **Possible values:** Any string
checkRequired	If set to true, the MCAD connector verifies that all PLM fields labeled as mandatory contain a value. If not, an error message is displayed to the user during save. **Possible values:** true | false
NonExistingPartsFromCAD	This option setting only affects property mapping. If set to "allowed", a property mapping towards the Item number can be used to create new Item objects, if the value mapped to the Item number does not identify an existing Item. If set to "remove", Item objects are not created in this case. **Possible values:** remove | allowed
AllowManualItemNumber	Makes the Item Number text field in the Details pane editable or

	non-editable. **Possible values:** true (editable) | false (non-editable)
VersionsToSearch	Defines how many older Design versions are searched if a "Where used" query is executed. Only retrieves Design versions which are older than the selected one. **Possible values:** Any integer value
VersionsToSearchFromCAD	**Possible values:** Any integer value
RelationsFromCAD	**Possible values:** true | false

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>
```


Table 28: ConnectionProperty Settings: Description

Setting	Purpose and available values
timeout	Defines the timeout value for web services responses from the PLM server in milliseconds. **Possible values:** Any number
bulksize	The server calls are divided into packages of this number of objects per call, depending on the server and network performance (use 20 for slower environments and 100 for faster ones). **Possible values:** 20 | 50 | 100
call-threads	Fallback value for the thread related option settings "call-threads-get", "call-threads-cuo-save", "call-threads-cuo-commit". If no value is given for one of these option settings, this value is used. **Possible values:** 1 | 2 | 4
call-threads-get	Number of concurrent threads for GET requests with which the EC services interface is called. **Possible values:** 1 | 2 | 4
call-threads-cuo-save	Number of concurrent threads for save requests with which the EC services interface is called. **Possible values:** 1 | 2

call-threads-cuo-commit	Number of concurrent threads for commit requests, structure and relationships creating requests, BOM requests with which the EC services interface is called. **Possible values:** 1 | 2
UploadThumbnails	UploadThumbnails = false | fileserver | ecs; fileserver – upload the CAD thumbnail from the client to the vault and attach the png to the Design in PLM. Note: In order to display the thumbnail in PLM configure AutoVue to: 1. NOT render the CAD native files for thumbnails 2. Render PNG files for thumbnails false – no thumbnail is uploaded from the CAD client to the server. ecs – hand over the thumbnail to Agile PLM via web service **Possible values:** false | fileserver | ecs

BrowserDisplay Section

```
<Structure>
        <Name>BrowserDisplay</Name>
        <FieldCollection>
        <Field><Name>HideSuccessSummary</Name><Value>false</Value></Field>
        <Field><Name>LazyLoad</Name><Value>true</Value></Field>
        <Field><Name>Tree.Menu.Zip</Name><Value>true</Value></Field>
        <Field><Name>AllowExistingChangesOnly</Name><Value>false</Value></Field>
        <Field><Name>DisableChangesAutonumbers</Name><Value>false</Value></Field>
        <Field><Name>ShowTreeView</Name><Value>true</Value></Field>
        <Field><Name>WorkspaceFolderClass</Name><Value>Design</Value></Field>
        <Field><Name>WorkspaceFolderAutonumber</Name><Value>Designs Number</Value></Field>
        <Field><Name>WorkspaceFolderDescription</Name><Value>host;path;file</Value></Field>
        <Field><Name>ShowFlyoutInBrowser</Name><Value>true</Value></Field>
        <Field><Name>ShowFlyoutInList</Name><Value>true</Value></Field>
        </FieldCollection>
</Structure>
```

Table 29: BrowserDisplay Section Settings: Description

Setting	Purpose and available values
HideSuccessSummary	If set to "true", the Save Summary dialog is only displayed for save processes where errors occurred. If set to "false", the Save Summary dialog is displayed at the end of each save process.

	Possible values: true | false
LazyLoad	Activates or de-activates the MCAD connector's LazyLoad feature. If LazyLoad is activated, PLM information in the MCAD GUI is only retrieved from PLM and displayed on user request (click the object to initiate the process). If it is de-activated, all PLM information for all objects visible in the MCAD GUI is retrieved and displayed as soon as a GUI widget appears on screen. Thus, activated LazyLoad may shorten onset times especially for a large number of PLM-known CAD objects. **Possible values:** true (activates LazyLoad) | false (de-activates LazyLoad)
Tree.Menu.Zip	Defines, if the "Zip and Upload Workspace" context menu item is visible in Workspace Manager. **Possible values:** true (show the “Zip and Upload Workspace” menu item) | false (hide the “Zip and Upload Workspace” menu item)
AllowExistingChangesOnly	If set to true, when manually entering a Change number into the Change number field, the Change assignment is only executed if the number of an already existing Change has been given. If set to false, entering a Change number into the Change number field causes the MCAD connector to create a new Change object, if none with the given number already exists. **Possible values:** true | false
DisableChangesAutoNumbers	Activates or de-activates the Change autonumber button (Save Preview window or Assign Change dialog window). **Possible values:** true (de-activates the button) | false (activates the button)
ShowTreeView	Defines the pre-set value for showing or hiding the tree view widget. This value only defines the default value of this option setting. The value is overwritten by the user-specific option setting in the $HOME/AgileCache/GUIConfig.xml configuration file. **Possible values:** true (pre-set value to show the Tree View widget) | false (pre-set value to hide the Tree View widget)
WorkspaceFolderClass	Defines the class of the PLM object that is used for uploading workspace folders via the MCAD connector's "Zip and Upload

	Workspace" functionality.
WorkspaceFolderAutoNumber	Defines the autonumber generator which is used to retrieve the Design number for the object that is used for uploading workspace folders via the MCAD connector's "Zip and Upload Workspace" functionality. Note: removing both entries causes the MCAD connector to create Design objects with a number derived from the current username and timestamp.
WorkspaceFolderDescription	
ShowFlyoutInBrowser	Activates or de-activates the thumbnail fly-out dialog in the MCAD connector's tree view widget. **Possible values:** true (activate the fly-out) | false (de-activate the fly-out)
ShowFlyoutInList	Activates or de-activates the thumbnail fly-out dialog in the MCAD connector's list view widget. **Possible values:** true (activate the fly-out) | false (de-activate the fly-out)

FlowControl Section

The FlowControl section controls behaviour during processes during the *Save* and *Load Preview* windows.

Table 30: FlowControl Section Settings: Description

Setting	Purpose and available values
cleanUnreserved	Reverts the save option to “do not save” for all CAD objects, that are not checked out by the current user, if the *Force User Checkout* option is activated. **Possible values:** true (revert the save option) | false (do not revert it)
AllowCheckoutOfNonLatest	Permits users to check out a Design version which is not the latest one in PLM via the MCAD connector. **Possible values:** true (check out of a non-latest Design version is allowed) | false (check out of a non-latest Design version is not allowed)
HelperPartAttachment	Defines, if attachments should be published for helper parts during the Publish to Item process. **Possible values:** true (publish attachments for helper parts) | false (do not publish attachments for helper parts)
HelperPartBOM	Defines, if the BOM should be published for helper parts during the Publish to Item process. **Possible values:** true (publish the BOM for helper parts) | false (do not publish the BOM for helper parts)
InstanceBOM	Defines, if the BOM should be published for instances of part families (configurations) during the Publish to Item process. **Possible values:** true (publish the BOM for part family instances) | false (do not publish the BOM for part family instances)
GenericBOM	Defines, if the BOM should be published for the generic of a part family during the Publish to Item process. **Possible values:** true (publish the BOM for the generic) | false (do not publish the BOM for the generic)

SuppressUpdateFromBOM	Suppresses the attribute update during BOM creation for existing child elements of a BOM structure (Title Block, Page Two, Page Three) during BOM creation. **Possible values:** true (suppress attribute update) | false (allow attribute update)
DrawingResolutionAction	Defines the *save option*, that is applied to any drawing Design object added to the Save Preview dialog using the *Save As confirmation popup*. The *Save As confirmation popup* appears every time a Save As is executed in the Save Preview, allowing users to have the MCAD connector perform a where-used query in Agile to add drawings related to the structure displayed in the Save Preview. **Possible values:** Save As | Check In | Save | Ignore; default: Save As

DateFormats Section

Defines a set of date and time formatting parameters based on the Java SimpleDateFormat specification. These formatting definitions can be applied to date and time values mapped between CAD and PLM.

```
<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>
```

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>IS_INCORPORATED</Name><Value>0</Value></Field>
                <Field><Name>DCO</Name><Value>0</Value></Field>
                <Field><Name>DCO_SEQUENCE</Name><Value>0</Value></Field>
                <Field><Name>DCO_STATUS</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>
```

Note: In order to make fields writable in the save preview the Value tag is set to 1 instead of 0.

Not all fields are supported to be editable while others are always editable. The following fields are always read-only, regardless of the corresponding Value tag:

- CAX_FIL_NAME
- CAX_FULL_NAME
- IS_INCORPORATED
- DCO
- DCO_SEQUENCE
- DCO_STATUS
- CHECKOUTUSER
- WORKFLOW_STATUS
- CAX_MODIFIED
- PLM_MODIFIED
- SAVED
- HAS_PRIVILEGE
- FILTER
- ASSIGNED
- CAX_MODEL_TYPE
- CAX_MODEL_REF
- CAX_LINK_TYPE
- CAX_LINK_REF
- CAX_TYPE

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, apart from the functionality of the “value” tags: Fields displayed in the Workspace Manager are never editable.

OverrideConfiguration Section

The MCAD connectors allow filtering the sub-classes and number generators available in the GUI based on CAD file types. These settings affect Create dialog, the Preferences window, the Save Preview window and the Save As override dialog. The sub-class and number generator filtering is configured based on a name-and-value XML syntax:

Name: A filter identifier (see *Table 31: OverrideConfiguration Section Settings: Description* below) optionally concatenated with the CAD file ending for the CAD files to which to apply the filter setting, e. g. `DisplayedDesignClassesSLDPRT` (file type specific filtering setting) or

`DisplayedDesignClasses` (not file type specific setting).

Value: Use “All” for not applying any filtering. Use a comma-separated list of the classes or number generators (display names) that should be displayed, e. g.: `Design,Construction,Prototype`

The Preferences dialog is not affected by file type specific filtering settings, except for the Drawing Item Subclass combo box.

File type specific filtering settings should be added to the *OverrideConfiguration* options group after the generic filtering settings. Refer to the example given below for a template on how a file type specific filtering setting might be configured.

```
<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>
```

Table 31: OverrideConfiguration Section Settings: Description

Filter Identifier	Purpose and available values
DisplayedDesignClasses	List of Design classes that display in UI **Possible values:** Comma separated list of classes (display names) or “All”.
DisplayedItemClasses	List of Item classes that display in UI **Possible values:** Comma separated list of classes (display names) or “All”.
DisplayedChangeClasses	List of Change classes that display in UI **Possible values:** Comma separated list of classes (display names) or “All”.
DisplayedDesignAutoNumber	List of Design AutoNumbers that display in UI **Possible values:** Comma separated list of autonumber generators (display names) or “All”.
DisplayedItemAutoNumber	List of Item AutoNumbers that display in UI **Possible values:** Comma separated list of autonumber generators

	(display names) or “All”.
DisplayedChangeAutoNumber	List of Change AutoNumbers that display in UI **Possible values:** Comma separated list of autonumber generators (display names) or “All”.

Example: File Ending Specific OverrideConfiguration Settings for SOLIDWORKS

```
<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>

        <Field><Name>DisplayedDesignClassesSLDPRT</Name><Value>DesignClass1</Value></Field>
        <Field><Name>DisplayedDesignClassesSLDASM</Name><Value>DesignClass1</Value></Field>
        <Field><Name>DisplayedDesignClassesSLDDRW</Name>
                <Value> DesignClass1,DesignClass2</Value></Field>
        <Field><Name>DisplayedItemClassesSLDPRT</Name><Value>ItemClass1</Value></Field>
        <Field><Name>DisplayedItemClassesSLDASM</Name><Value>ItemClass2</Value></Field>
        <Field><Name>DisplayedItemClassesSLDDRW</Name>
                <Value>ItemClass1,ItemClass2</Value></Field>
</FieldCollection>
</Structure>
```

FileNaming Section

```
<Structure>
    <!-- common file naming section in PLM file attachments -->
    <Name>FileNaming</Name>
    <FieldCollection>
        <!--Field><Name>PDF</Name><Value>%NUMBER%_%REV%.PDF</Value></Field -->
        <Field><Name>PDF</Name><Value>%Item.NUMBER%_%Item.REV%.PDF</Value></Field>
        <Field><Name>STP</Name><Value>%Item.NUMBER%_%Item.REV%.STP</Value></Field>
        <Field><Name>JT</Name><Value>%NUMBER%_%REV%.JT</Value></Field>
        <Field><Name>PNG</Name><Value>thumbnail.png</Value></Field>
    </FieldCollection>
</Structure>
```

The FileNaming section provides an easy way to rename files on transfer into PLM. This should be done with viewable files only. Depending on the file extension different rules are possible. For each file extension only one rule is supported. The file naming rule is parsed during the file transfer process and can contain values from the Design object and from the Item. Refer to the class configuration which CAX-NAMES and field idents can be used.

Sample for Design Values: %NUMBER%_%REV%.JT -> combines Design Number and Version.

Sample for Item Values: %Item.NUMBER%_%Item.REV%.JT -> combines Item Number and Revision.

Sample for Fix Values: thumbnail.png -> combines Item Number and Revision.

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>
```

Table 32: Viewables Section Settings: Description

Setting	Purpose and available values
ViewablesDrawing	Visible viewable file types for drawing files. **Possible values:** List of file endings (use semi-colons as list separator).
ViewablesModel	Visible viewable file types for 3D files. **Possible values:** List of file endings (use semi-colons as list separator).

WorkspaceDeleteViewables Section

Defines the viewable file types (file endings) the MCAD connector's "Delete Viewables" functionality deletes from a workspace folder.

Table 33: Viewables Section Settings: Description

Setting	Purpose and available values
ViewablesDrawingExtensions	File formats for drawings, which are recognized by the *Delete Viewables* function in the Workspace Manager, and thus, deleted by this function.
ViewablesModelExtensions	File formats for 3D models, which are recognized by the *Delete Viewables* function in the Workspace Manager, and thus, deleted by this function.

PartFamilies Section

Table 34: PartFamilies Section Settings: Description

Setting	Purpose and available values
FamilySelection	If set to true, operations (e. g. Check Out) applied to a component of a part family (configuration) through the MCAD connector are implicitly applied to all other components of the same part family. **Possible values:** true | false
FamilyInstanceNumbering	Defines the numbering schema for Design objects of instances of part families (configurations). GENERIC_INDEX – Instances get the same Design number as the part family generic assigned, appended with a numerical index. GENERIC_CONFIG – Instances get the same Design number as the part family generic assigned, appended with the CAD specific name of the instance. false – Instances get independent Design numbers assigned. **Possible values:** GENERIC_INDEX | GENERIC_CONFIG | false **Important:** This option setting is not supported for the MCAD connector for Solid Edge.
CountDelimiter	Separator character used for appending part family (configurations) related appendices (refer to FamilyInstanceNumbering) or indices for multiple drawings which are related to the same model (refer to DrawingNumberFromModel).
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 Item object corresponding to the generic Design gets a number with dash assigned (like P001-000, instances get numbers like P001-001,… assigned) false - the Item object corresponding to the generic Design does not get a number with dash assigned (like P001, instance get numbers

	like P001-001,… assigned)
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>
                <Field><Name>DesignRevisionLogic</Name><Value>editable,publish,noparentheses</Valu
                e></Field>
                <Field><Name>AssignChangeCheckout</Name><Value>false</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>DrawingNumberFromModel</Name><Value>true</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>
                <Field><Name>PublishPartSite</Name><Value>none</Value></Field>
        </FieldCollection>
</Structure>
```

Table 35: ChangeProperties Section Settings: Description

Setting	Purpose and available values
InitialRevision	Initial Item revision value
RevisionSequence	Defines the proposed Revision values the MCAD connector implicitly assigns to Change objects on Change creation. **Possible values:** List of Revision values (strings) separated by comma
RevisionSequenceEditor	Defines, if the MCAD connector provides a combo box (drop down list) for editing the Item Revision columns. The entries of the combo box are taken from the option setting "RevisionSequence". If set to false and if the Revision field is editable, users are allowed to enter free text into the Revision field. **Possible values:** true | false

SetLifecyclePhase	Defines, if the MCAD connector provides a combo box (drop down list) for editing the Lifecylce Phase column. If set to false and if the Lifecylce Phase field is editable, users are allowed to enter free text. **Possible values:** true | false
DesignRevisionLogic	Comma-separated list of the following values. If set the value controls the following behavior: 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**:**editable,publish,noparentheses**
InitialDesignRevision	If no Item object is assigned to a Design, this value defines the "Item Revision" string displayed for the versions of this Design object. **Possible values:** Any string
DesignRevisionForIntroductory	Defines the string, that replaces the "Introductory" Item Revision in the Design version drop-down list. By changing this option setting, it is possible to use other string values for representing the Introductory Revision. **Possible values:** Any string
AssignChangeCheckout	If true, this triggers an implicit check out of the related Design object when assigning a Change object. Does not work if "ForceUserCheckout" is set to "true". **Possible values:** true | false
VersionSeparator	Separator character used for separating the major and minor Item Revision displayed in the Design version drop-down list. Has no effect, if the minor version is de-activated.
PartDesignNumbering	If set to true, on Item number retrieval (e. g. when pressing the Item autonumber button), the MCAD connector implicitly assigns the retrieved Item number to the Design number of the related Design object. **Possible values:** true | false

DrawingNumberFromModel	Defines the numbering schema for Design objects of CAD drawings: If set to true, the Design number of the drawing's Design object gets a number equal to the Design number of the underlying model, appended with a numerical index. **Possible values:** true | false
PushPartRevisionToDesign	If set to true, when changing the Item Revision in the MCAD connector's GUI, the changed Revision is automatically transferred to the Item Revision displayed in the Design version drop-down list. **Possible values:** true | false
PushDesignRevisionToPart	If set to true, when changing the Design version in the MCAD connector's GUI, the changed version is automatically transferred to the Item Revision. **Possible values:** true | false
PublishAttachments	Defines the attachment types, identified by their file ending, that are attached to an Item object when executing a publish process. **Possible values:** any | Comma separated list of file endings (without “.”)
PublishAttachmentType	The value in this option setting is written to the "Attachment Type" PLM field when executing a publish process. If set to "NONE", nothing is written to that field. **Possible values:** none | Any string except “none”
PublishIntroductory	If set to true, the MCAD connector also executes publish processes for Items without assigned Change objects (means for Items with "Introductory" Item Revision). **Possible values:** true | false
PublishPartSite	Defines the "Site" to which BOM changes are written when executing a publish process, if "Sites" are enabled in Agile. Use the "value" none if "Sites" are disabled. **Possible values:** none | Any string expect “none”
MaxDesignNumberLength	Defines the maximum allowed length of the Design number field values. Any Design number entries longer than the specified length are cut off by the MCAD connector.

	Possible values: Any integer
AllowedNumberCharacters	Defines a set of characters which may be used for Item and Design numbers. The MCAD connector replaces any character not contained within this list with underscores ("_") in the Item and Design number fields. Use the value "ANY" to disable this functionality. This functionality does not affect autonumber assignement, autonumbers retrieved from Agile are never changed by the MCAD connector. **Possible values:** ANY | Set of characters without delimiters, like: `abcdefghijklmnopqrstuvwxyz1234567890`
Change_Category	Defines the default *Change Category* value for Item Change Orders created by the MCAD connector. Default: none **Possible values:** Any string except “none”, which exists in the drop-down list of the Change Category attribute for Item Changes in Agile.
PublishErrorHandling	Defines the behaviour of the MCAD connector if errors during the publish process occur. If set to "ERROR", the save, check in and publish process is completely cancelled if an error occurs. If set to "WARNING", an error report is displayed in the Save Summary window and only the publish process is cancelled, not the save or check in. **Possible values:** ERROR | WARNING
PublishIntegerQuantities	If set to true, the MCAD connector transfers quantity values (Design structure, BOM) in integer representation instead of a number with a decimal place. **Possible values:** true | false

READONLY_FF_FIELDS Section

This XML structure defines all File Folder PLM fields which should not be updated by the MCAD connector (read-only fields), defined by their field ID. Though the XML definition is based on a name-and-value syntax, the value field itself is not evaluated by the connector.

Important: The fields already defined in this section should not be removed from CaxConfig.

READONLY_ITEM_FIELDS Section

This XML structure defines all Item fields which should not be updated by the MCAD connector (read-only fields), defined by their field ID. Though the XML definition is based on a name-and-value syntax, the value field itself is not evaluated by the connector.

CAX_NAMES_BY_ID Section

Allows executing an attribute mapping during save as operations. Refer to *Mapping Values on Save As* for details.

Login Access Administration in PLM

This section provides detailed information about all necessary steps to enable login restriction for defined client versions that connect to the PLM server.

All steps need to be executed as an administrator user that has rights to modify roles and privileges.

1. Create a new FileFolder subclass, for example “MCAD-Access”. This subclass should not have an AutoNumber assigned as the numbers need to be entered manually. This could also be a Designs subclass, but for separation purposes and easier configuration, a FileFolder subclass is recommended.
2. Find any existing Discovery privileges for File Folders (for example Discover All File Folders) and modify the criteria to exclude the new File Folder class created in step #1. This prevents any non-admin user to modify the access setup. It might be required to create a new criteria because the existing one could not be modified.
3. Create a new object in subclass “MCAD-Access”, for example “MCAD-3.2.0.0.130701“, for each version you want the user(s) to grant access to. The new object number needs to be prefixed with “MCAD-“ and the version number is exactly the same as displayed in the Web Components status line in the bottom of the dialogs.
4. Create a new privilege for “Discovery”, for example “Connect with MCAD 3.2.0.0” with criteria “Discover MCAD 3.2.0.0” on object type “MCAD-Access” and the condition “MCAD-Access Title Block.Number Equal To MCAD-3.2.0.0.130701”.
5. Add this privilege to the user’s CAD specific role, for example “CAD Designer”.
6. Remove the privilege “Discover All File Folders” from that same role. This is vital as otherwise this privilege overwrites any restrictions created in step #4.
7. More specific privileges on certain file folders could be added to the user’s role if needed.
8. To secure the access to the “MCAD-Access” subclass, any write privileges for non-admin users should be revoked on any object of that subclass.

Important note:

To be backwards compatible or to allow customers to use MCAD without any MCAD-access setup, the client is connected successfully if the MCAD-access subclass or the version-specific FileFolder does not exist.

The latter requires having the appropriate MCAD-Access FileFolder created before an MCAD version, other than the ones already set up in the system, is passed to the user. For any MCAD version for which the access should be rejected, a corresponding FileFolder object needs to be created, but no discover privilege should be assigned to the user’s role for this particular object.

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 is open, 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.

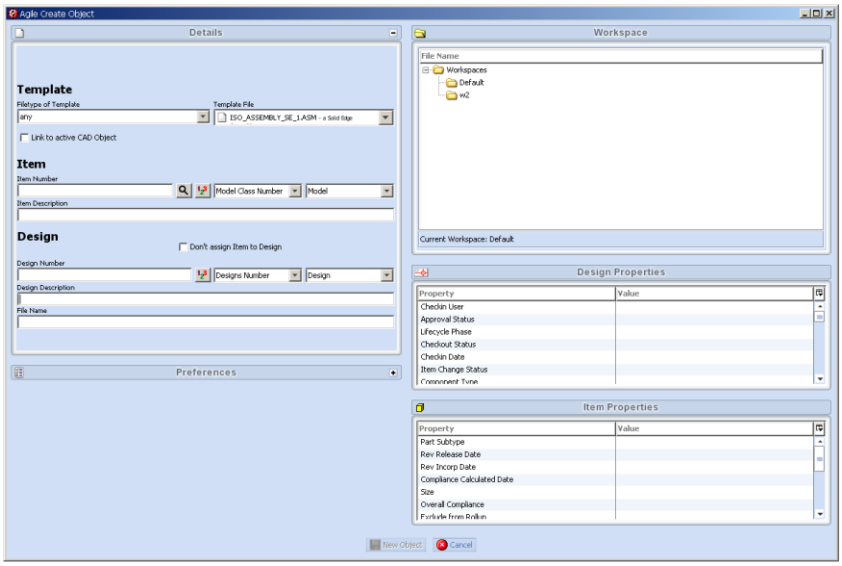


Figure 16: Create Object frame, used to create new objects

Creating the Template Structure in Agile

The Create Object dialog uses a certain data structure 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 uppercase 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. For example 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 vice versa. The %SITE% is assigned by editing the file **<Install Directory>\ini\CAXConfig.xml**. In this case **<Install Directory>** refers to the MCAD connector's installation folder. To 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. Additionally, it is necessary to switch off forced AutoNumber usage when creating new Design objects. This can be achieved with the following steps:

- Log in 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, for example MCAD-START-CATIA. Click *Save*, the template structure has been created and can be loaded with CAD files now.

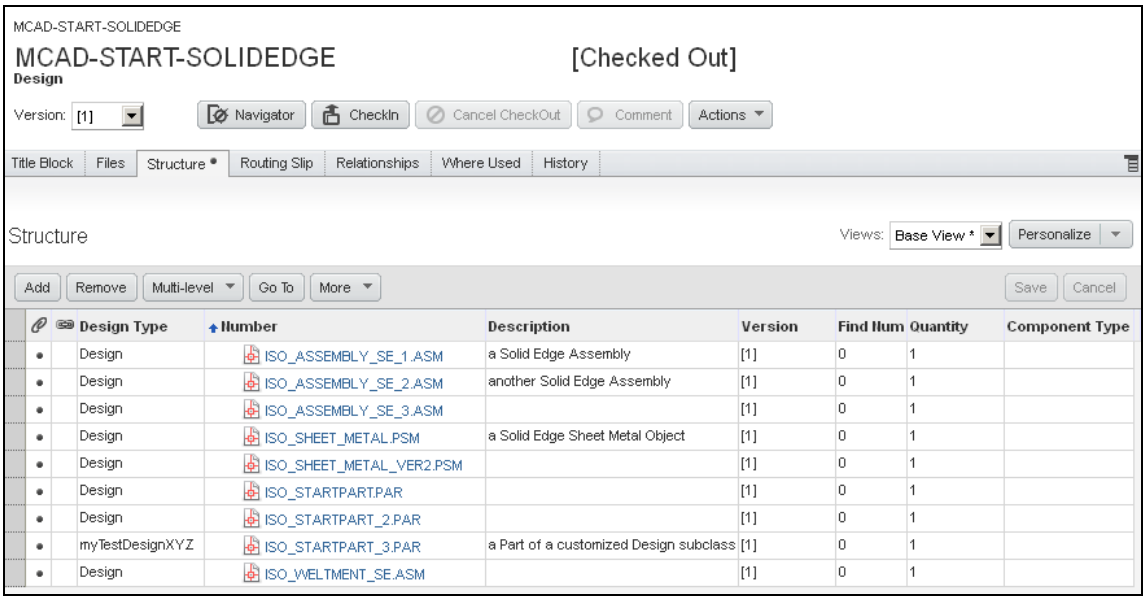
Adding Template Files to the Structure

A template file is a normal CAD file that is made available in 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 that 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 becomes 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 uppercase (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 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 similar to the following:



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 17: Example template structure for Solid Edge

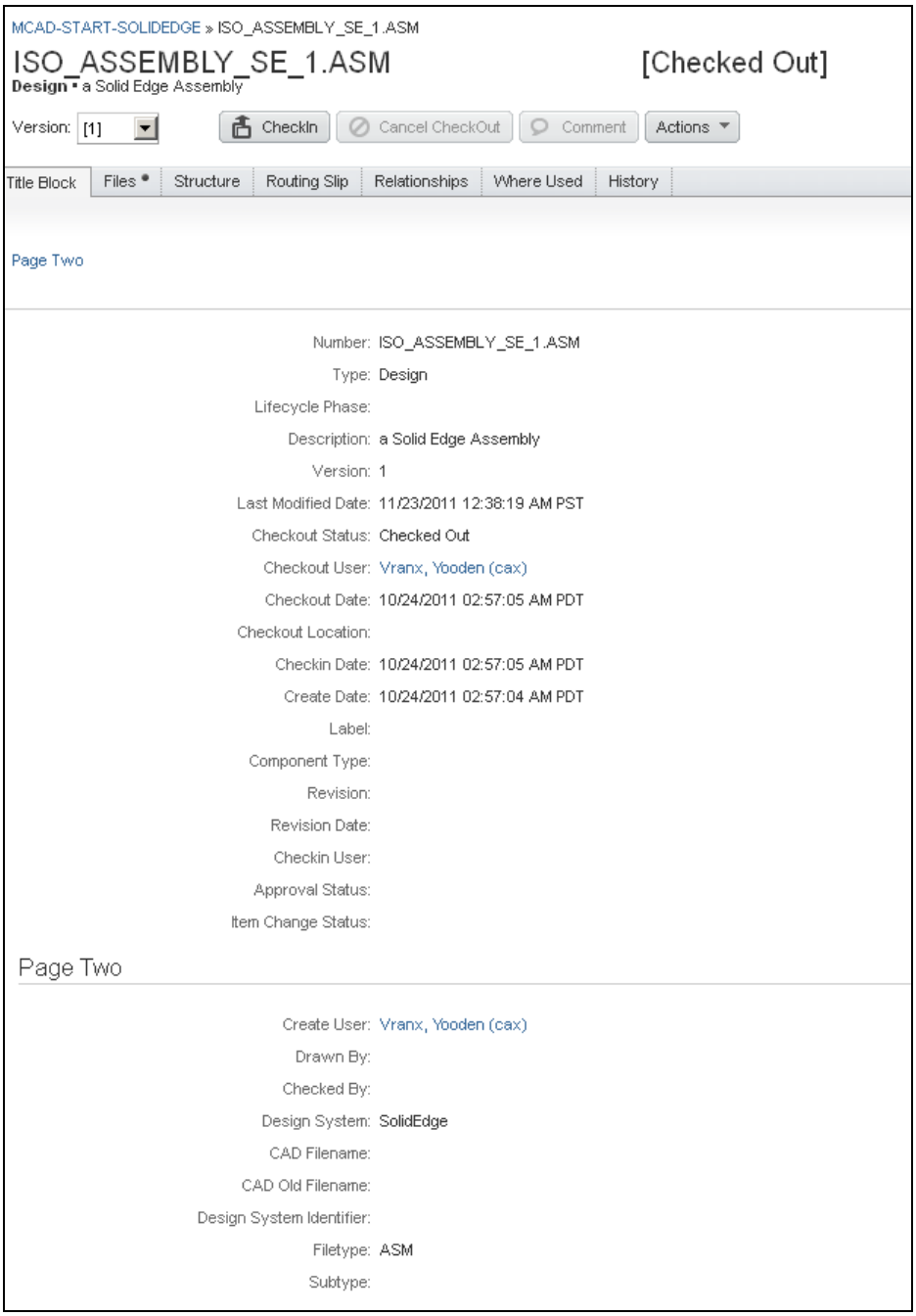


Figure 18: Example of a Title Block of a template object

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 by 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, for example 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.

Structure Resolution

The Create functionality of MCAD does not perform any kind of structure resolution for the elements of the template structure. Due to this, on template download, only the actual CAD file selected in the Create dialog is downloaded and opened in the CAD system regardless of its structure. This also accounts for Creo format files. Due to this, format files should be managed outside of the Agile and the MCAD connector. For details, refer chapter *Managing Format Files* above.

Possible Errors

This section gives a summary of 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. There are two types of error messages: Critical messages, which are caused by errors that do not allow a safe continuation of program execution, and information messages are caused by improper configurations or missing data. Such errors do not cause serious danger for program execution. However, in most cases the user cannot continue the action that caused the message.

Table 36: 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 respond 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 (Creo), Configurations (SOLIDWORKS), 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

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.

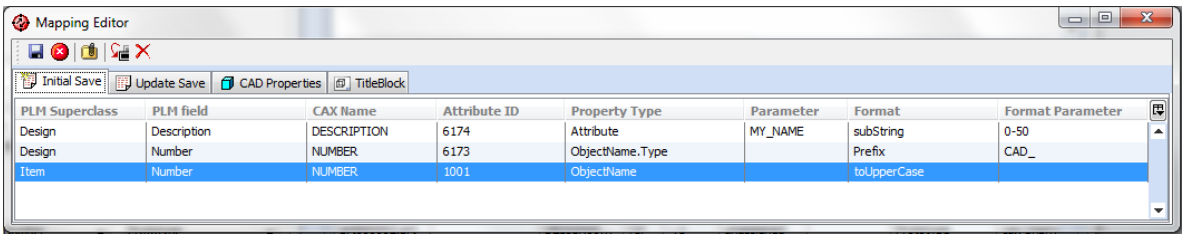


Figure 19: Mapping Editor

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.

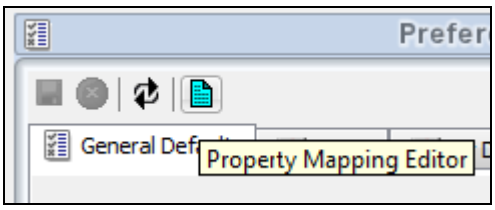


Figure 20: Mapping Editor – Button in the preferences pane

The toolbar of the Mapping Editor has the following functionality.

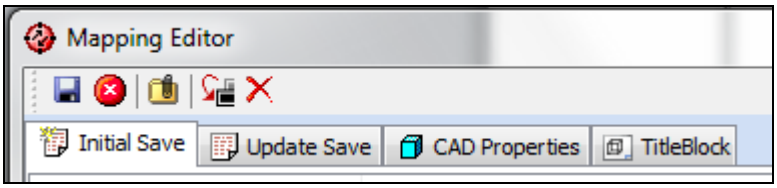


Figure 21: Mapping Editor

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 is 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.

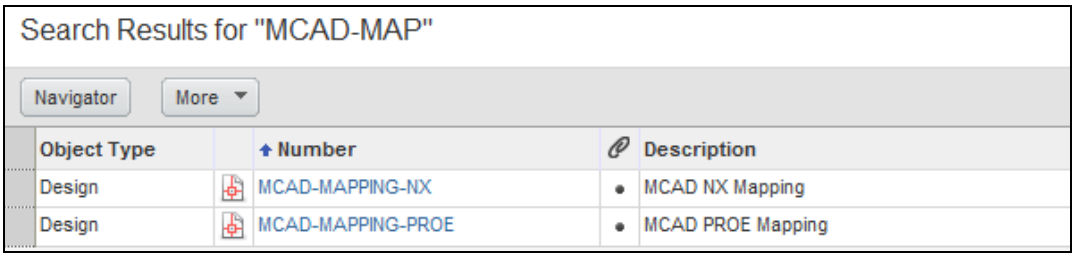


Figure 22: Agile 9: Search Results for "MCAD-MAP"

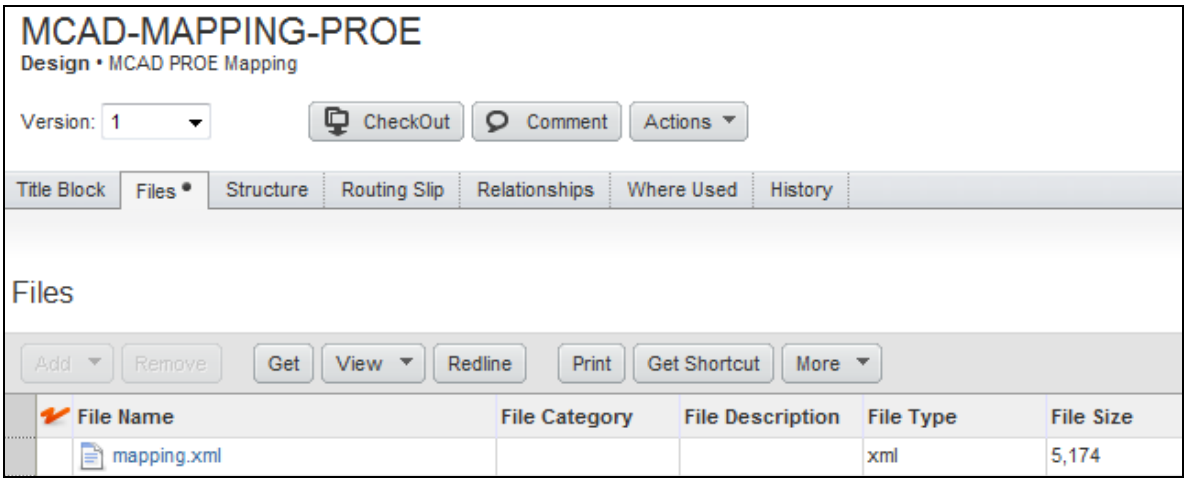


Figure 23: Agile 9: MCAD-Mapping-PROE

Mapping CAD properties to PLM fields

The first two tabs define the mappings of CAD properties to PLM fields. For CAD objects that are not 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, except for the MCAD connector for Creo Parametric: In this integration, the *Update Save* tab is also executed for objects that are not yet known to PLM. 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*.

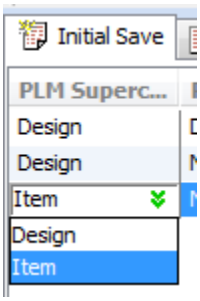


Figure 24: Initial Save Options

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.

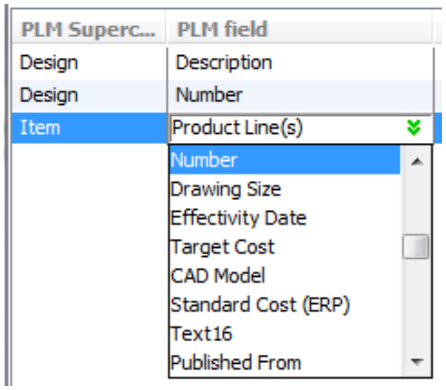


Figure 25: Available Agile 9 PLM field for Item

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

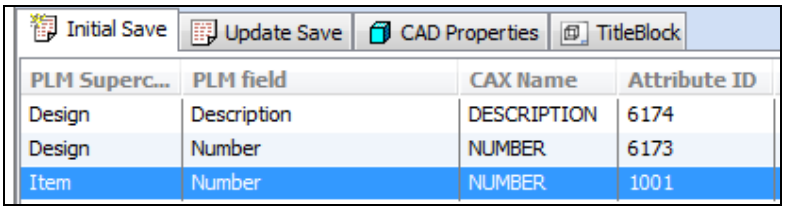


Figure 26: Initial Save Mapping for Item

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.

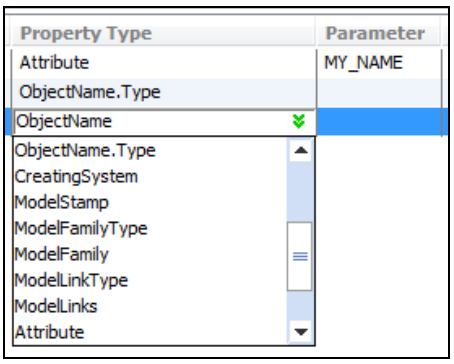


Figure 27: Browsing for available CAD Properties

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.

Table 37: Available Default CAD Properties

Property Type	Purpose
ModelFullName	File name including path
ModelName	File name without path
ModelExtension	File extension
ModelType	File type, equal to the CAD file extension in most CAD systems
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 (Creo)
DrawingModel	Model assigned to the drawing (Creo)
DrawingModelName	Model name assigned to the drawing (Creo)
DrawingModelType	Model extension assigned to the drawing (Creo)
HelperPartIdent	Helper part property

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 SOLIDWORKS Summary Information Properties

The MCAD connector does not support mapping properties from the *Summary* tab of the *Summary Information* dialog in SOLIDWORKS to Agile. Those properties include *Author*, *Keywords*, *Comments*, *Title* and *Subject*. Mapping properties listed in the *Custom* and *Configuration Specific* tabs works.

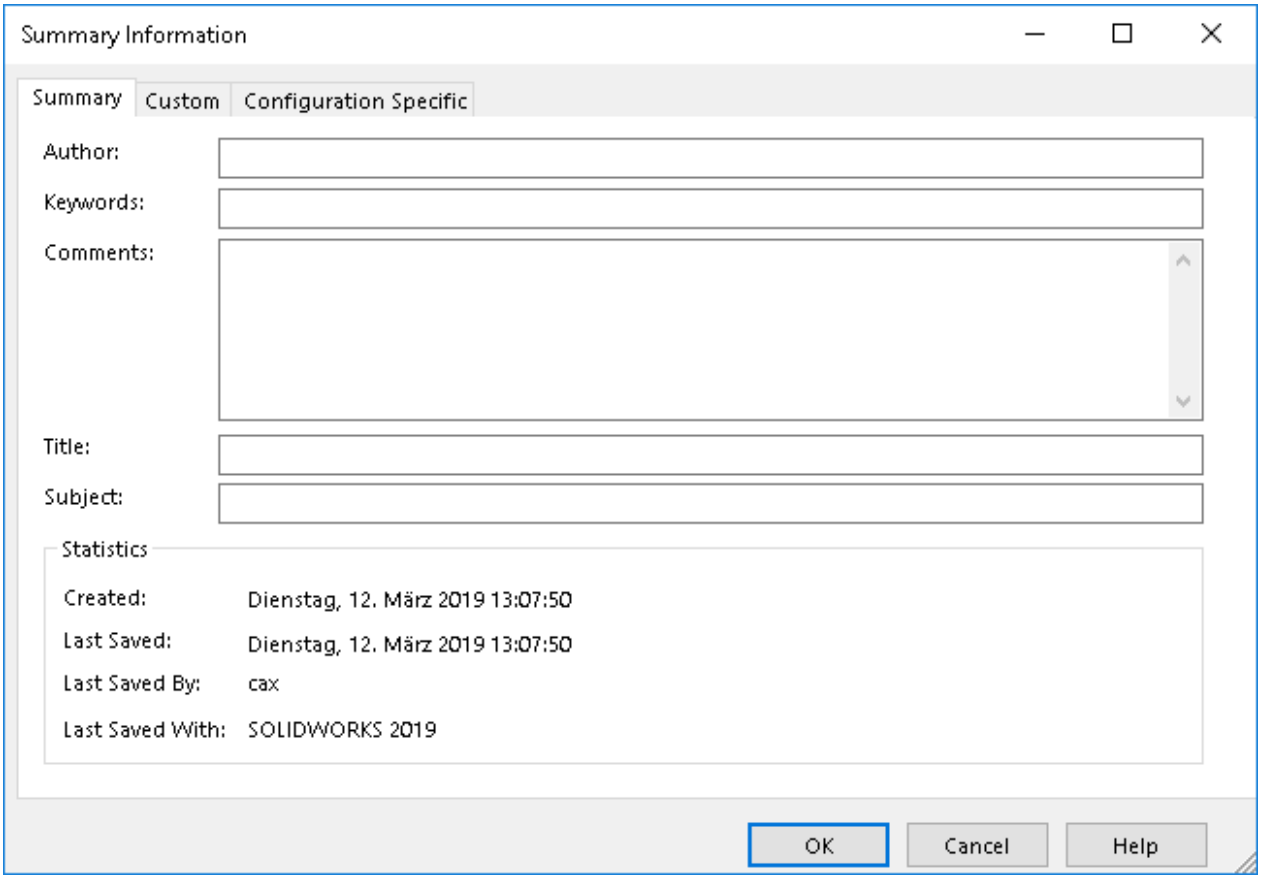


Figure 28: Summary tab of the Summary Information dialog

Mapping Empty Values to and from PLM Fields

Since version 3.6.1, the MCAD connector allows mapping empty values to and from PLM attributes. Every attribute that should be allowed to contain empty values needs to be added to one of the following white lists in CaxConfig.xml:

For objects of the *File Folders* base-class or any sub-class thereof:

FF_WHITE_SPACE_LIST

For objects of the *Items* base-class or any sub-class thereof:

ITEM_WHITE_SPACE_LIST

For objects of the *Changes* base-class or any sub-class thereof:

CHANGE_WHITE_SPACE_LIST

Each of the white lists represents an XML *Structure* object containing a list of *Fields*. The value **1** assigned to a *Field's Value* indicates that empty values are allowed for the corresponding PLM attribute. The value **0** means that empty values are not allowed. If an attribute is not listed in the white list, the MCAD connector implicitly assumes that empty values are not allowed as well.

The *Name* tag of the XML *Field* must contain the *Base ID* of the PLM attribute in question.

Example:

```
<Structure>
    <Name>FF_WHITE_SPACE_LIST</Name>
    <FieldCollection>
        <Field><Name>6174</Name><Value>1</Value></Field>   <!-- empty values allowed -->
        <Field><Name>1303</Name><Value>0</Value></Field>   <!-- empty values not allowed -->
    </FieldCollection>
</Structure>
```

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.

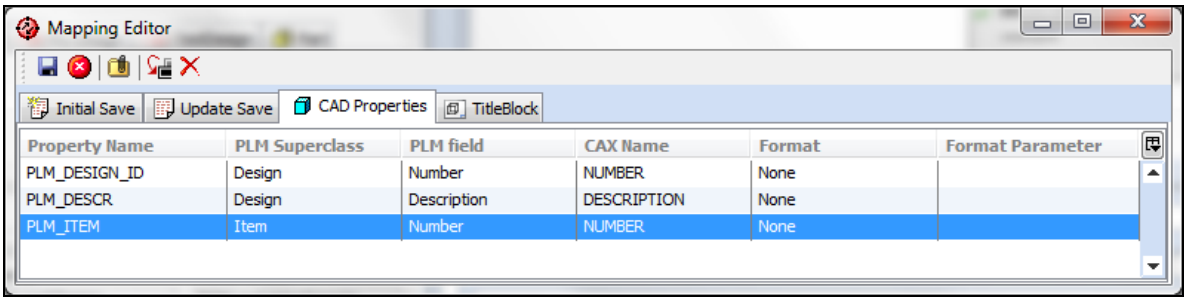


Figure 29: Mapping PLM Values to CAD Properties: CAD Properties Tab

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.

Table 38: Internal CAD Property Names and their purpose

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.

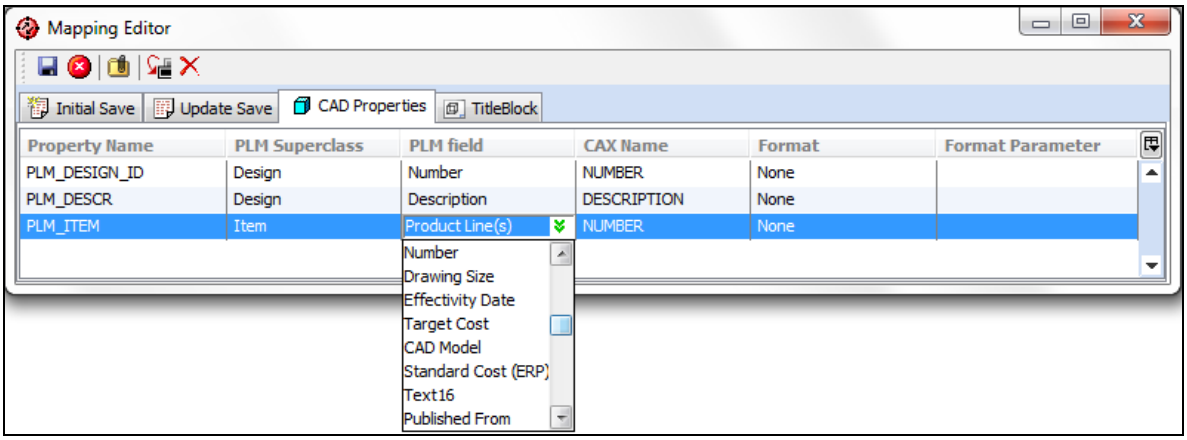


Figure 30: Mapping Editor: CAD Properties: PLM Field: Combobox

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.

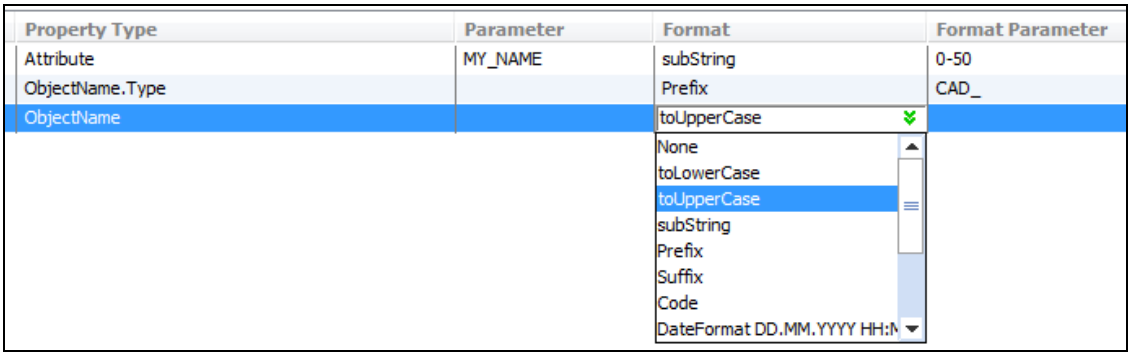


Figure 31: Format Parameter during the mapping

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.

Table 39: Format Parameter during the mapping: Description

Format	Purpose
None	No formatting action is executed
toLowerCase	Convert the value to lower case characters
toUpperCase	Convert the value to uppercase 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

Mapping Values on Save As

In MCAD release 3.5.0.0, the functionality for customizing attribute assignments in the MCAD connector has been removed from the product (Preferences dialog, sub-class tabs). This was done to prevent incompatibilities to the default MCAD data model which could be introduced by improper configurations. However, the attribute customization also allowed customers to extend the data model

of MCAD to their particular needs, e. g. by configuring an attribute to which MCAD would write the source object of a *save as* process. Since MCAD 3.5.0.0, this functionality was not available any longer.

To restore the previously available functionality regarding the save as source object, a new mapping functionality was introduced to the MCAD connector. This mapping functionality allows users to define attributes to which the source objects of a save as process should be written during the execution of that process. The actual mapping definition is done in CaxConfig.xml using an option structure with the following XML syntax:

```
<Structure>
<Name>CAX_NAMES_BY_ID</Name>
        <FieldCollection>
                <Field><Name>ObjectType.AttributeID</Name><Value>CAX_COPY_FROM</Value></Field>
                <Field><Name>ObjectType.AttributeID</Name><Value>CAX_COPY_FROM</Value></Field>
        </FieldCollection>
</Structure>
```

`ObjectType` is the type of object to which the value should be written. This can be `FF` (FileFolder or Design) or `Item` (any kind of Item). `AttributeID` is the attribute ID of the target PLM attribute. The actual function value CAX_COPY_FROM defines the source object of an MCAD save as process. For Design objects this is the Design number, for Items the Item number.

Example configuration:

```
<Structure>
<Name>CAX_NAMES_BY_ID</Name>
        <FieldCollection>
                <Field><Name>FF.2471842</Name><Value>CAX_COPY_FROM</Value></Field>
                <Field><Name>Item.1301</Name><Value>CAX_COPY_FROM</Value></Field>
        </FieldCollection>
</Structure>
```

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 Add-in in CAD (CAD menus and icons) must be configured CAD-specific.

Table 40: CAD Connector Components

CAD Tool	How to configure Add-in Language
SOLIDWORKS	The CAD Add-in language is configured in **components\xml\XPlmConnector.xml** in this setting: `<Language>EN</Language>`. Valid values are `EN`, `FR`, `DE`, `CN`.
Solid Edge	The CAD Add-in language is configured in **components/xml/XPlmConnector.xml** in this setting: `<Language>EN</Language>`. Valid values are `EN`, `FR`, `DE`, `CN`.
Creo Parametric	The CAD Add-in language is configured in **xACP.cfg** with the `AcpLang=English` setting. Valid values are `English`, `French`, `German, Chinese_cn`.

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

CAD Thumbnail Support

Inside the Web Components thumbnails are extracted from the CAD native file. The CAD system stores thumbnail views into CAD binary data that are extracted using the same routines like the Windows Explorer is using. The minimum requirement is a Windows 7 or higher system. On Windows XP thumbnails are not supported.

EC does not generate the thumbnail content. The dialogs extract the thumbnail from the CAD file. The thumbnail content is provided by the CAD vendors. Please refer to the documentation of your CAD tool how to enable thumbnail generation in the CAD files. Some CAD tools like Creo need additional viewing components installed or enabled (Solidworks settings) on the local CAD client machine to provide this feature. The connector extracts the same thumbnail visible in the thumbnail view of Windows Explorer.

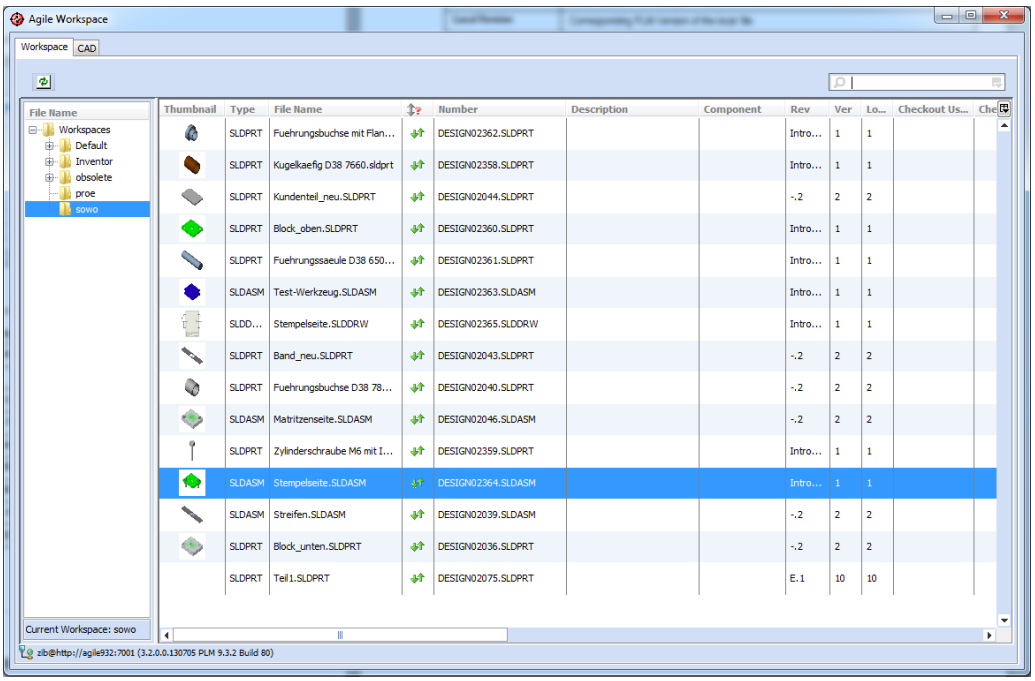


Figure 32: Agile Workspace

The thumbnails are held locally only and are shown inside the Workspace Manager in small icons. A bigger flyout is shown on the small icon, on a node in the browser views or on the filename column in the table view.

Transfer CAD thumbnails to PLM

The thumbnail transfer is controlled by the UploadThumbnails option setting in CAXConfig.xml. If set to *fileserver*, CAD thumbnails are transferred to PLM by uploading them to the file vault. A preview.png file is visible in the Files tab of the CAD Design object in this case. If set to *ecs*, an API function is used to store the thumbnails in the PLM database to prevent file transfer overhead. This is the default value. If set to *false*, the thumbnail transfer to PLM is switched off.

AutoVue configuration for CAD thumbnails

In order to show the CAD thumbnail in the Web Client you should disable the generation of thumbnails in AutoVue for the CAD native file formats and enable the png file rendering in AutoVue only. Otherwise the AutoVue generated thumbnails can overwrite the CAD thumbnail. Since the CAD assembly is not loaded for thumbnail generation this reduces the load on the AutoVue server and database.

Thumbnails in the Load Preview

The thumbnails in the load preview use the PLM thumbnails if available, and download the file from the vault on demand (on a show flyout request). If local files and thumbnails are available on the local disk already, then the thumbnail file is not downloaded from PLM and the local representation is used.

Troubleshooting

Certain versions of SOLIDWORKS showed problems regarding thumbnail generation and display. Should generic file type icons be displayed instead of actual thumbnails, we recommend to update to the latest SOLIDWORKS service pack. For SOLIDWORKS 2015 we recommend SP 4.0.

Legacy Workspaces

Workspace folders created with MCAD 3.3 or earlier versions are not supported by MCAD 3.4 or later, due to the file caching mechanism being changed in MCAD 3.4 (individual .clb files instead of the workspace-specific cache.xml file). Workspaces created from MCAD 3.3 (or earlier) thus need to be checked in before migrating to MCAD 3.4 (or later) and checked out again after the migration.

Toolbox and Library Parts (ACW and ACE Only)

Since release 3.6, the MCAD connector for SOLIDWORKS supports the *Design Library Toolbox* and since release 3.6.2 the Solid Edge connector also supports it. CAD engineers can now use toolbox parts in SOLIDWORKS and Solid Edge transparently.

Configuring the Standard Parts Directory

For using the toolbox/library parts feature, the MCAD connector needs to be configured for the directory where the toolbox/library parts are stored:

For SolidWorks:

XPlmSolidWorksConnector.xml, "SolidWorksStandardPartDir" option setting, recommended default value: **C:\SolidWorks Data**

For Solid Edge:

XPlmSolidEdgeConnector.xml, "SolidEdgeStandardPartDir" option setting, recommended default value: **C:\Program Files\Solid Edge** *XY*

Note: *XY* is the version of Solid Edge in this case, e. g. ST 10.

The recommended default values given above provide the directories where the CAD systems in question usually store their toolbox/library parts. Using these directories would be preferable when intending to use the default toolbox and library parts supplied with the CAD systems. However, the MCAD connector can support any other directory, thus the configuration can be adapted to support toolbox and library part directories in any other place on the file system as long as the CAD system supports it.

Saving

Toolbox/library parts are saved to Design objects in Agile (as usual). The **Design.PageTwo.Name Format** attribute shows "TOOLBOX" for these objects.

Loading

Toolbox/library parts are loaded the same way as any other CAD object. They appear in the active MCAD workspace and have .clb files assigned to them. Toolbox parts loaded to the active CAD workspace do not cause any conflicts related to a separate toolbox parts directory outside of the workspace that might exist, since both CAD files (the toolbox part file in the local MCAD workspace and the toolbox part file in the standard parts directory) link to the same Design object in Agile.

Renaming

Toolbox parts never get renamed by the MCAD connector. The corresponding *Renaming on Save* and when *Renaming on Load* functions have no effect for toolbox and library parts.

Viewables

Viewables are not created for toolbox/library parts. Regardless of the viewable related option settings selected in the Preferences dialog.

Files Without Collaboration Files (.clb Files)

When saving library/toolbox parts to Agile, the MCAD connector queries the PLM system for the file name of the toolbox part. If the query returns a result, the MCAD connector re-assigns the corresponding Design object to the toolbox part in question. This way, toolbox/library parts that had been saved to Agile before can be linked to their Design objects even without having a .clb file assigned.

This functionality also allows several users that utilize the same set of library/toolbox parts (but not on a shared location, like a network drive) to work with toolbox parts without having to worry about conflicts regarding the concurrent usage of those files.

Save As

Once saved to Agile and thus PLM-known, toolbox/library parts are not affected by the MCAD connector's Save As function. This means, that no new Design number is assigned to them in case that they are selected and the Save As button is clicked.

Strict CAD Modification Workflow (ACW and ACE only)

Since release 3.6, the MCAD connectors for SOLIDWORKS and Solid Edge support a strict CAD modification workflow based on Agile checkouts. If enabled, this workflow causes the MCAD connector to not allow users to modify PLM-known CAD objects unless they are checked out in Agile.

Option Settings

The strict CAD modification workflow feature is controlled by a number of options settings that are explained in detail in the following table.

Table 41: Overview on the option settings

Option Setting	Function

`%CAD%_ReserveBeforeModify_EnableUseCase`	Global switch to activate or deactivate the the strict CAD modification workflow feature. Possible values: 0 (deactivated) and 1 (activated)
`%CAD%_ReserveBeforeModify_NotifyUser`	Enables or disables popup dialogs related to the strict CAD modification workflow feature. The actual dialog displayed also depends on the %CAD%_ReserveBeforeModify_AutoReserve option setting. Possible values: 0 (deactivated) and 1 (activated)
`%CAD%_ReserveBeforeModify_AutoReserve`	Enables or disables automatic check out attempts related to the strict CAD modification workflow feature. Possible values: 0 (deactivated) and 1 (activated)

%CAD% is the place holder for the CAD system, `SolidWorks` or `SolidEdge` in this case.

Depending on the values of the option settings given in Table 41, the behaviour of the feature changes according to the following table.

Table 42: Valid option values and combinations and resulting behaviour

Option setting	Behaviour				
	A	B	C	D	
`%CAD%_ReserveBeforeModify_EnableUseCase`	1	1	1	1	0
`%CAD%_ReserveBeforeModify_NotifyUser`	1	0	1	0	0 or 1
`%CAD%_ReserveBeforeModify_AutoReserve`	1	1	0	0	0 or 1

Behaviour A: The MCAD connector displays popup dialog for all file(s) that are not checked out. In the popup dialog, users can opt to check out the file(s) in question. This removes the read-only flag.

Behaviour B: The MCAD connector implicitly tries to check out the file(s) in question. If successful, the read-only flag is removed.

Behaviour C: The MCAD connector checks if the file(s) in question are checked out by the user. If so, the read only flag is removed, if not, an information popup is displayed.

Behaviour D: The strict CAD modification workflow feature is deactivated.

Activating the Feature

The strict CAD modification workflow feature is deactivated per default. It can be activated as follows:

For SOLIDWORKS

In XPlmSolidWorksA9Connector.xml (under ...\XPLM Solution GmbH\OraclePLM\xml), modify below settings as given:

SolidWorks_ReserveBeforeModify_EnableUseCase = 1
SolidWorks_ReserveBeforeModify_NotifyUser = 0 or 1 (*)
SolidWorks_ReserveBeforeModify_AutoReserve = 0 or 1 (*)

In XPlmSolidWorksConnector.xml (under ...\XPLM Solution GmbH\OraclePLM\xml), modify below settings as follows:

SolidWorksEvent_PartModifyNotify = true
SolidWorksEvent_AssemblyModifyNotify = true
SolidWorksEvent_DrawingModifyNotify = true

* For details on which setting combination is valid and what the resulting behaviour is, refer to Table 42.

For Solid Edge

In XPlmSolidEdgeA9Connector.xml (under ...\XPLM Solution GmbH\OraclePLM\xml), modify below settings as follows:

SolidEdge_ReserveBeforeModify_EnableUseCase = 1
SolidEdge_ReserveBeforeModify_NotifyUser = 0 or 1 (*)
SolidEdge_ReserveBeforeModify_AutoReserve = 0 or 1 (*)

In XPlmSolidEdgeConnector.xml (under ...\XPLM Solution GmbH\OraclePLM\xml), modify below setting as follows:

SolidEdgeEvent_AfterActiveDocumentChange = true

* For details on which setting combination is valid and what the resulting behaviour is, refer to Table 42.

Changes to the CAD Workflow

If activated, the strict CAD modification workflow feature introduces a number of changes to how users work with the MCAD connector. The feature models the check-out status of an object in Agile to the corresponding CAD object in SOLIDWORKS or Solid Edge and prevents users from making or saving modifications to non-checked out objects. To prevent this, the MCAD connector activates the CAD system's read-only mode implicitly whenever users open or load an Agile managed CAD object that is not checked out for them.

Due to some differences in the implementation of the read-only modes for SOLIDWORKS and Solid Edge, the actual behaviour of the CAD system may be different:

SOLIDWORKS: Whenever users attempt to use a feature that would modify the CAD data, the following popup dialog is displayed. The dialog allows users to directly attempt to check out the object in question.

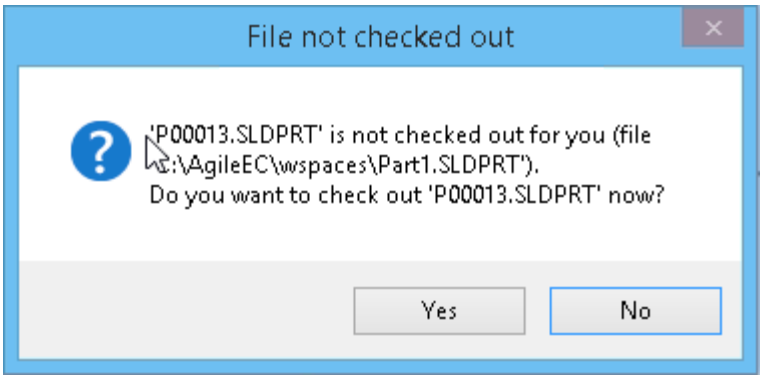


Figure 33: Popup dialog of the ACW integration for non-checked out objects.

Solid Edge: When opening a CAD object that is not checked out or attempting to make the object active (happens when switching to that object's tab in Solid Edge or when loading the object from disk), the following popup dialog is displayed. The dialog allows users to directly attempt to check out the object in question. Note that Solid Edge does not allow users to use the "Save" buttons for read-only CAD objects.

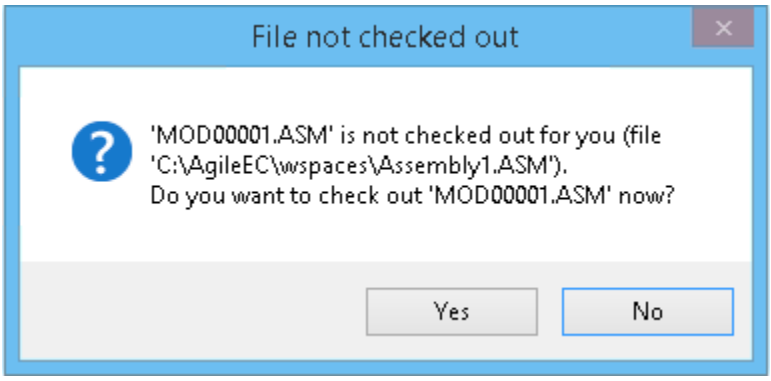


Figure 34: Popup dialog of the ACE integration for non-checked out objects.

Required Server Configuration for Using MCAD Connectors

This section provides a complete summary of configuration options required for the MCAD 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 MCAD 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 connector. You may need to enable additional fields according to the desired 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. **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.**

Large text attributes are currently not supported.

Activating Thumbnails

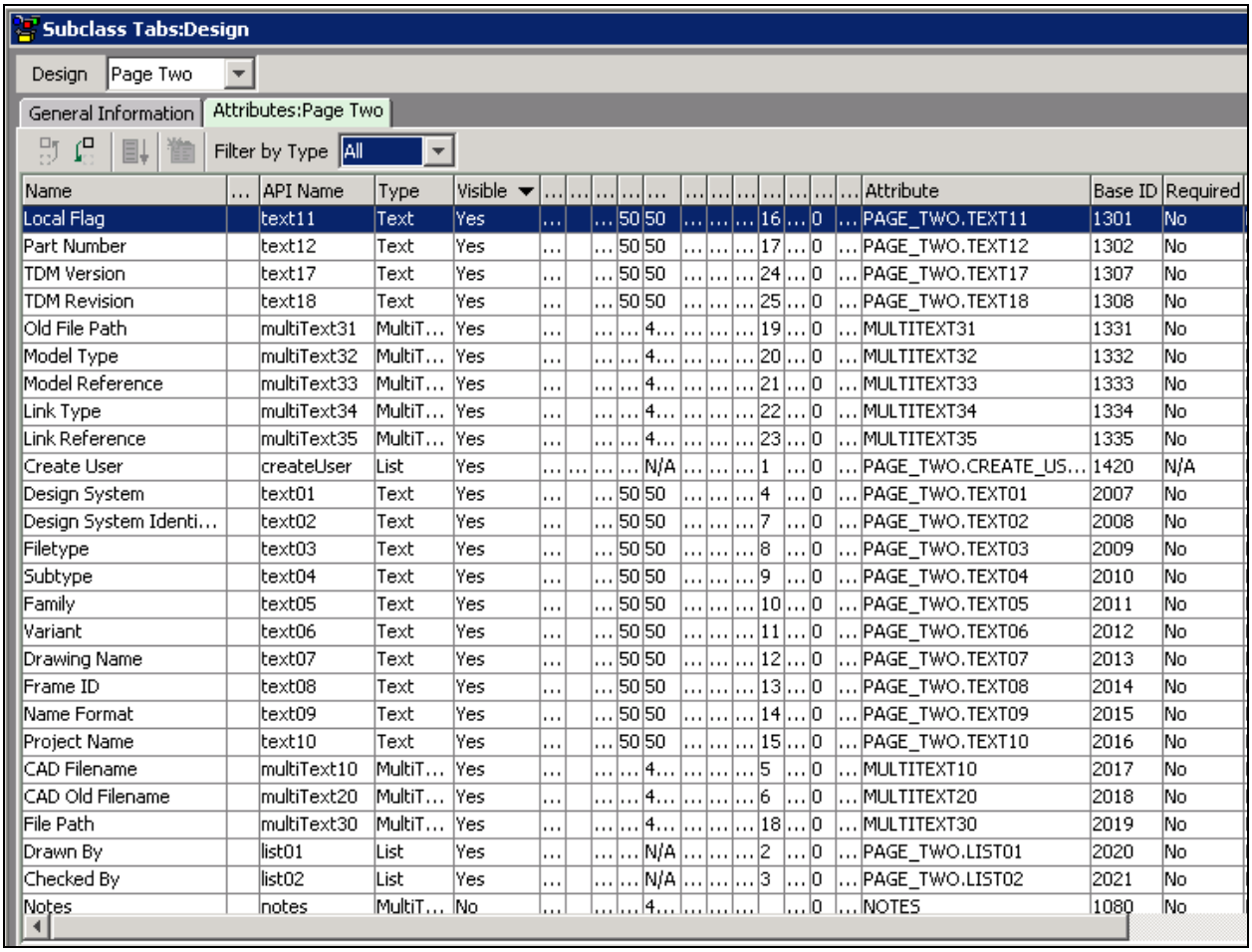
The MCAD connectors support the thumbnail functionality of Agile PLM. Therefore, activate the following server side option settings:

- Navigate to Admin → Server Settings → Preferences → General Information.
- *Enable* the option settings “Thumbnail Display” and “Thumbnail and Streaming File Pre-Generation”.

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

Figure 35: Agile 9: Admin Client: Subclass Tabs: Design: Attributes: Page Two

Designs – Files

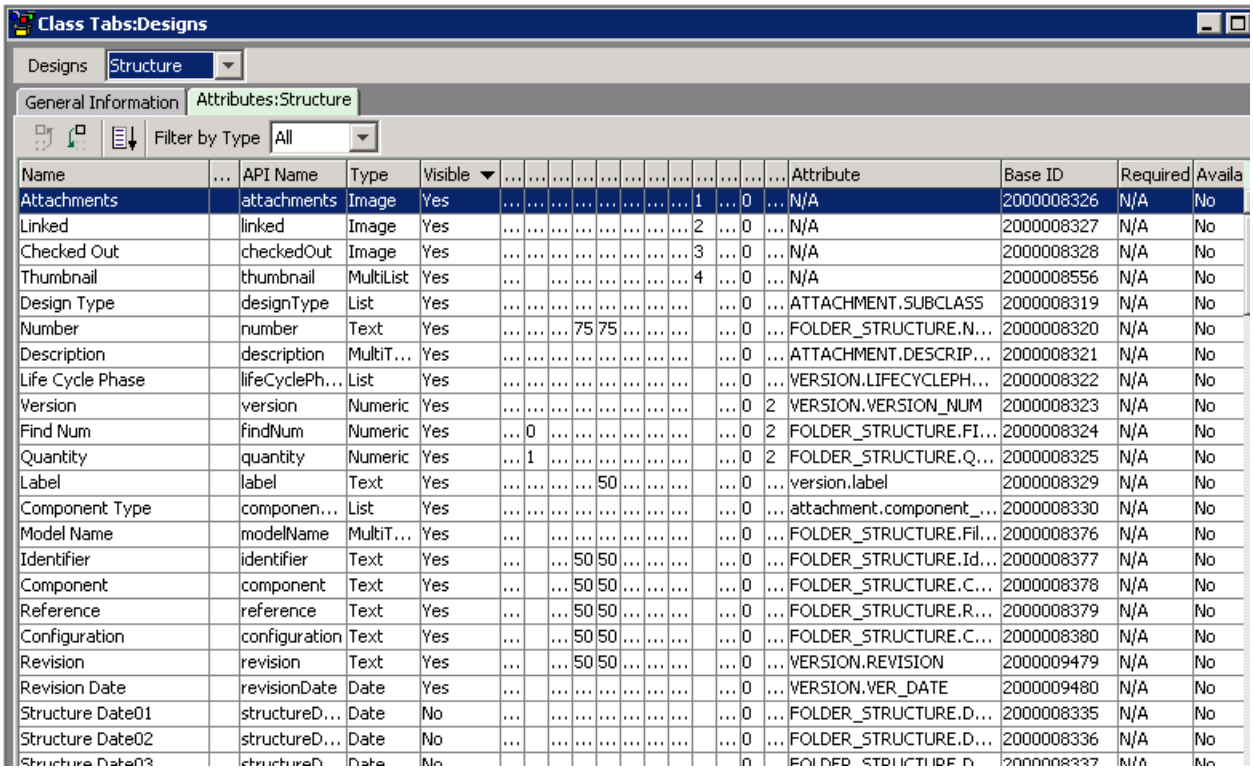
Make sure the following fields are enabled:

- 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

Figure 36: Agile 9: Admin Client: Subclass Tabs: Design: Attributes: Structure

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)

Class Tabs:Designs

Designs | Where Used

General Information | Attributes:Where Used - Attachment | Attributes:Where Used - Design

Filter by Type: All

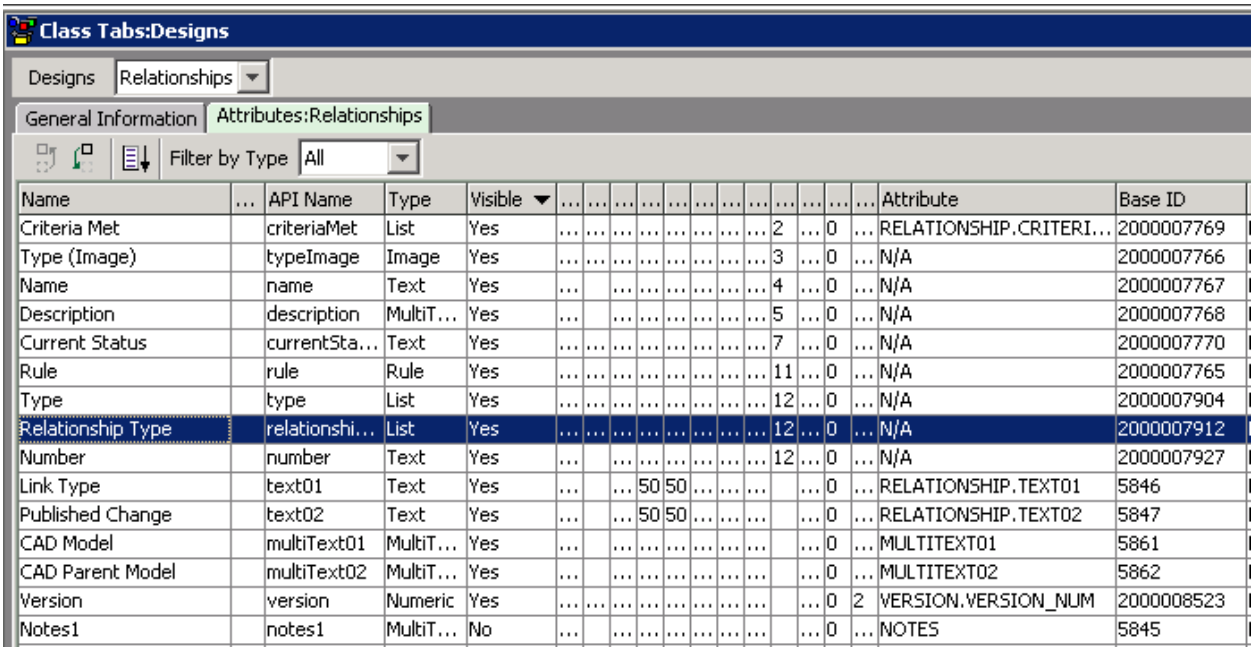
Name	API Name	Type	Visible	Attribute	Base ID	Required
Thumbnail	thumbnail	MultiList	Yes	N/A	2000008558	N/A
Design Type	designType	List	Yes	ATTACHMENT.SUBCLASS	2000008497	N/A
Number	number	Text	Yes	FOLDER_STRUCTURE.N...	2000008498	N/A
Description	description	MultiT...	Yes	ATTACHMENT.DESCRIP...	2000008499	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

Figure 37: Agile 9: Admin Client: Subclass Tabs: Design: Attributes: Where Used - Design

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

Figure 38: Agile 9: Admin Client: Subclass Tabs: Design: Attributes: Relationships

Parts – Title Block

Make sure the following fields are configured as follows:

- Number (ID: 1001)
    - Include Characters: All
    - MaxLength: 75

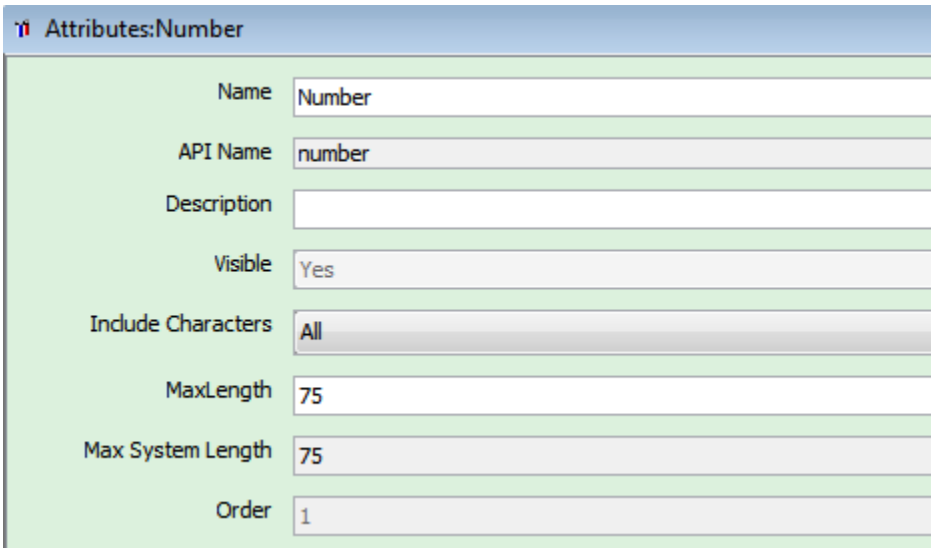


Figure 39: Item Number Attribute Configuration

Parts – BOM

Make sure the following fields are enabled:

- Item Number (ID: 1011)
- 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
- Item Number (ID:1011) - not the standard attribute name, must be modified manually

Class Tabs:Parts

Parts BOM

General Information | Attributes:BOM

Filter by Type All

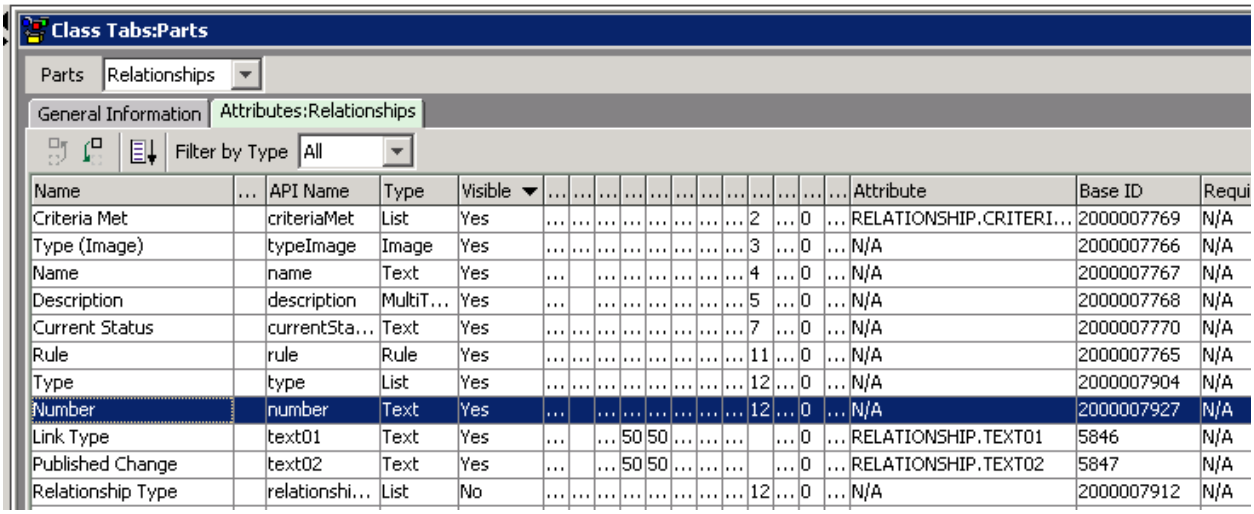
Name	...	API Name	Type	Visible	...	...	...	...	...	...	...	...	...	...	...	...	Attribute	Base ID	R
Attachments (Image)		attachmen...	Image	Yes	...	...	...	...	...	...	...	...	1	...	0	...	N/A	12630	N
Manufacturer (Image)		manufactu...	Image	Yes	...	...	...	...	...	...	...	...	2	...	0	...	N/A	12631	N
Pending Changes (Im...		pendingCh...	Image	Yes	...	...	...	...	...	...	...	...	3	...	0	...	N/A	12632	N
Has Been Redlined (I...		hasBeenR...	Image	Yes	...	...	...	...	...	...	...	...	4	...	0	...	N/A	6795	N
Has Quality(Image)		hasQuality...	Image	Yes	...	...	...	...	...	...	...	...	5	...	0	...	N/A	7954	N
Price (Image)		priceImage	Image	Yes	...	...	...	...	...	...	...	...	6	...	0	...	N/A	2000007815	N
Pending Declarations ...		pendingDe...	Image	Yes	...	...	...	...	...	...	...	...	7	...	0	...	N/A	2000011130	N
Thumbnail		thumbnail	MultiList	Yes	...		...	...	...	...	...	...	8	...	0	...	N/A	2000008550	N
Item Number		itemNumber	Text	Yes	...	...	...	75	75	...	...	...	9	...	0	...	BOM.ITEM_NUMBER	1011	N
Item Description		itemDescri...	MultiT...	Yes	...	...	...	...	...	...	...	...	10	...	0	...	ITEM.DESCRIPTION	1020	N
Item Rev		itemRev	Text	Yes	...	...	...	...	20	...	...	...	11	...	0	...	REV.REV_NUMBER	1021	N
BOM Quantity		qty	Text	Yes	...	1	...	20	20	...	...	...	12	...	0	...	BOM.QUANTITY	1035	N
Min Qty		minQty	Numeric	Yes	...		...	...	...	...	...	...	13	...	0	2	BOM.MINIMUM_NUMBER	2000008542	N
Max Qty		maxQty	Numeric	Yes	...		...	...	...	...	...	...	14	...	0	2	BOM.MAXIMUM_NUMBER	2000008543	N
Find Num		findNum	Text	Yes	...	0	...	5	8	...	...	...	15	...	0	...	BOM.FIND_NUMBER	1012	N
Optional		optional	List	Yes	...		...	...	...	...	...	...	16	...	0	...	BOM.IS_OPTIONAL	2000008540	N
Mut Excl		mutExcl	List	Yes	...		...	...	...	...	...	...	17	...	0	...	BOM.IS_MUTUALLY_EXC...	2000008541	N
Ref Des		refDes	Text	Yes	...	...	...	...	...	...	...	...	18	...	0	...	REFDESIG.LABEL	1019	N
Sites		sites	List	Yes	...		...	...	...	...	...	...	19	...	0	...	BOM.SITE	12205	N
BOM Notes		BOMNotes	MultiT...	Yes	...		...	...	...	...	...	...	20	...	0	...	BOM.NOTES	1036	N
Summary Compliance		summaryC...	List	Yes	...	...	...	...	...	...	...	...	21	...	0	...	REV.COMPLIANCY	2000011100	N
CAD Filename		BOMMultiT...	MultiT...	Yes	...		...	...	...	...	...	...		...	0	...	MULTITEXT30	1341	N
Identifier		BOMText01	Text	Yes	...		...	50	50	...	...	...		...	0	...	BOM.TEXT01	2175	N
Component		BOMText02	Text	Yes	...		...	50	50	...	...	...		...	0	...	BOM.TEXT02	2176	N
Reference		BOMText03	Text	Yes	...		...	50	50	...	...	...		...	0	...	BOM.TEXT03	2177	N
BOM Description		BOMDescri...	MultiT...	No	...		...	...	...	...	...	...		...	0	...	BOM.DESCRIPTION	1013	N

Figure 40: Agile 9: Admin Client: Subclass Tabs: Parts: Attributes: BOM

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

Figure 41: Agile 9: Admin Client: Subclass Tabs: Parts: Attributes: Relationships

Parts – Changes – Attributes: Pending Changes

Make sure the following fields are enabled:

- Lifecycle Phase (ID: 2000009590)
- Workflow (ID: 2000009595)
- Change Category (ID: 2156)

Configuration when using Documents classes as Items for Drawings:

Documents – 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)

Documents – Pending Changes

Make sure the following fields are enabled:

- Lifecycle Phase (ID: 2000009590)
- Workflow (ID: 2000009595)
- Change Category (ID: 2156)

Documents – BOM

Since BOM are rather uncommon for Documents classes, usually no configuration of the BOM tab is required. However, if BOM should be used, the following configuration needs to be implemented.

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

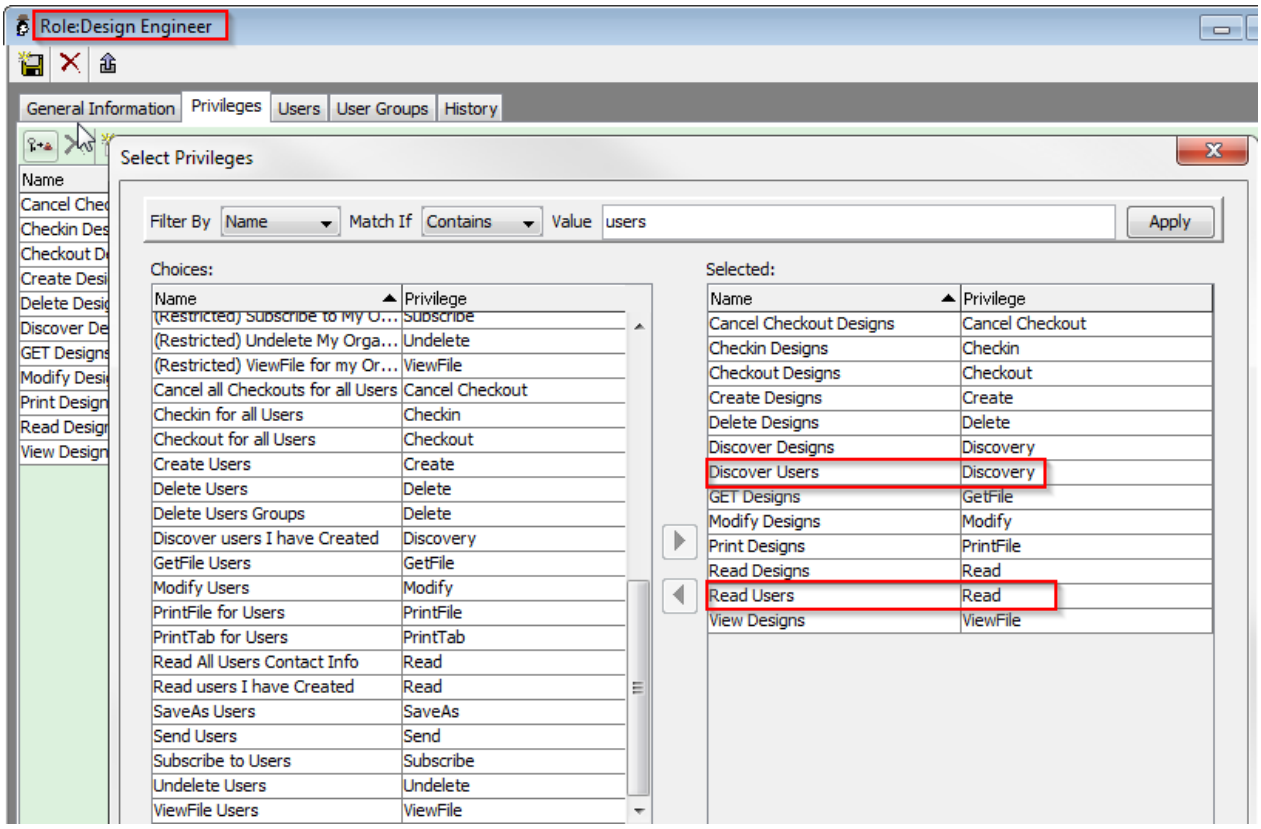
Duplicate Find Numbers

To use this feature (refer to the User Guide) it is necessary to set the “DuplicateFindNumbers” option to “Allow” under SystemSettings -> SmartRules.

Allowing MCAD Engineers to See Checkout Users

Per default, Agile PLM is configured to not allow non-administrator users to see the content of the Checkout User attribute within the Designs class. It is recommended to allow all MCAD users to read and discover this attribute to avoid conflicting access to Design objects. Once privileges are granted, MCAD users can see which other user is currently working on a Design object (means, having checked out the Design).

In order to activate read-level access to the Checkout User attribute (Design, TitleBlock), add the privileges Read Users and Discover Users to the main MCAD engineer role (usually this is Design Engineer).


Role:Design Engineer
General Information
Privileges
Users
User Groups
History
Select Privileges
Filter By
Name
Match If
Contains
Value
users
Apply
Choices:
Name
Privilege
(Restricted) Subscribe to My O...
Subscribe
(Restricted) Undelete My Orga...
Undelete
(Restricted) ViewFile for my Or...
ViewFile
Cancel all Checkouts for all Users
Cancel Checkout
Checkin for all Users
Checkin
Checkout for all Users
Checkout
Create Users
Create
Delete Users
Delete
Delete Users Groups
Delete
Discover users I have Created
Discovery
GetFile Users
GetFile
Modify Users
Modify
PrintFile for Users
PrintFile
PrintTab for Users
PrintTab
Read All Users Contact Info
Read
Read users I have Created
Read
SaveAs Users
SaveAs
Send Users
Send
Subscribe to Users
Subscribe
Undelete Users
Undelete
ViewFile Users
ViewFile
Selected:
Name
Privilege
Cancel Checkout Designs
Cancel Checkout
Checkin Designs
Checkin
Checkout Designs
Checkout
Create Designs
Create
Delete Designs
Delete
Discover Designs
Discovery
Discover Users
Discovery
GET Designs
GetFile
Modify Designs
Modify
Print Designs
PrintFile
Read Designs
Read
Read Users
Read
View Designs
ViewFile

Enabling HTTPS for Engineering Collaboration Clients

Introduction

If the MCAD connectors or custom client applications 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.

To make the client side work properly with the HTTPS server, you must import the server's Certificate Authority (CA) certificate into a Java Keystore (JKS) for the clients and a client certificate needs to be generated with the help of this CA certificate.

This document covers both one-way authentication and two-way authentication (also called mutual authentication), but the latter one is recommended as it is considered as most secure.

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.

Exporting the server certificate

When you want to use HTTPS with the Agile PLM application server, there should already be a server side 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) in the next 2 chapters and start with step 2. Please only do step 1 if you would like to create self-signed certificates for testing purposes. Self-signed certificates should not be used in production environments.

Creating the keystore for the server

When there is no JKS file available on the server, create a keystore for the server by executing the 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?
```

Exporting the server certificate

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

Certificate stored in file <plm-server.crt>
```

Creating the Client Keystore for One-Way Authentication

Creating 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 the 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?
```

Importing the server's certificate into the client keystore

For this step, copy the exported server certificate file “plm-server.crt” to the client side to import it into the client keystore.

```
# 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
```

Creating the Client Keystore for Two-Way Authentication

Creating the keystore for the client

Follow the same steps as in the previous section.

Importing the server's certificate into the client keystore

Follow the same steps as in the previous section.

Exporting the client certificate

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

Certificate stored in file <plm-client.crt>
```

Importing the client certificate into the server keystore

```
# 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:
```

The file “plm-server.jks” needs to be moved to or updated in the same location on the server where it resided before and where the WebLogic application server is configured to access it. It is highly recommended to back this JKS file before making any updates.

Configuring the MCAD Connectors for HTTPS

To make use of the keystore client file, certain Java startup parameters need to be added to the MCAD’s JVM startup call (defined in 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.

Copy “plm-client.jks” to 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=Agile123
-Djavax.net.ssl.trustStoreType=jks
```

The setting of “javax.net.ssl.trustStoreType” is optional as JKS is the default store type.

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 …
```

Configuring custom client applications for HTTPS

To make use of the certificates in the keystore and the client file, certain Java system properties needs to be added into the custom client code. This enables to Java application to validate the client certificate with the server’s one and then establish an HTTPS trusted and secure connection.

The files “plm-client.jks” should be copied into the client application’s directory, but could essentially reside anywhere on the local file system..

The keystore password is the one that had been used with the keytool command above.

The code below shows how to use the required properties for the Java programming language, but this should works similar on any other programming language, for example C#.

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

System.setProperty(“javax.net.ssl.trustStorePassword”, “Agile123”);
```

In Java, the value for the property “javax.net.ssl.trustStoreType” does not need to be set as it is “jks” by default. But it might be necessary in other programming languages.

This also works with other programming languages that could be used to generate the client stubs for the web services. Please consult the corresponding documentation for the details.

Enabling WSS for Engineering Collaboration Clients

Introduction

Since Oracle Agile PLM 9.3.4, Web Service Security (WSS) is supported. This is a core feature of the WebLogic application server to allow setting policies for web service clients to access web services with certain authentication methods only. For further information, please see the corresponding documentation for your WebLogic version on http://docs.oracle.com.

For the list of supported policies for Oracle Agile PLM; please see the Oracle Agile PLM Administration Guide.

For any web services client with enabled web service security, only the policy "oracle/wss_username_token_over_ssl_client_policy" is supported. As this policy requires SSL, the WebLogic application server must be configured to serve HTTPS requests.

Configuring the MCAD Connectors for WSS

The MCAD connectors automatically detect if WSS is enabled for the Core Services and EC Services and use the required client side policy.

As WSS requires HTTPS to be used, the steps in the previous chapter on how to enable HTTPS for MCAD need to be performed.

For WSS, it is usually only required to use One-Way Authentication and not Two-Way Authentication (called Mutual Authentication in the previous chapter).

With One-Way Authentication the client would only need to import the server certificate if the server certificate is self-signed. If the server certificate had been issued by a Certificate Authority (CA) so that it is considered as trusted, this is not needed.

Intermediate Certificates

There might be Certificate Authorities (CAs) that issue a couple of *intermediate chain certificates* instead of just one server certificate (e. g. three certificates for the certificate authority *GoDaddy*, refer the following link for additional explanations https://uk.godaddy.com/help/what-is-an-intermediate-certificate-868).

This is mainly the case when the certificate authority wants to hide or protect their root certificate and thus provides intermediate certificates instead. In this case, it is required to import all of the certificates contained in the CRT file provided into the *Java Key Store* (JKS) file. It may also be the case that several CRT files are provided by the CA; in this case all of the CRT files need to be imported.

Troubleshooting

This chapter provides information on known issues related to Agile and the MCAD connector that users might encounter as well as aids in troubleshooting.

Lifecycle Phase Value Does not Appear for Document Objects

Affected Agile versions: Agile 9.3.5, Agile 9.3.6

Under certain circumstances Agile does not allow the MCAD connector to write the Lifecycle Phase values for Documents sub-classes to the corresponding attributes. This is a server side error. In order to circumvent this error, a new functionality has been implemented for the MCAD connector that allows writing the Lifecycle Phase value to a Design attribute, in addition to the usual functionality of writing the Lifecycle Phase to the Affected Items tab of a Change object. The Lifecycle Phase value can be transferred back from the Design attribute in question to the Affected Items tab using a process extension (PX) in Agile. The aforementioned server error appears not to affect PXes.

For the MCAD connector, the additional Lifecycle Phase mapping can be activated in CaxConfig.xml by adding the following XML structure to the file:

```
<Structure>
       <Name>WORKAROUNDS</Name>
       <FieldCollection>
       <Field><Name>DuplicateItemLifecycleToDesign</Name><Value>true</Value></Field>
       </FieldCollection>
</Structure>
```

Once activated, the MCAD connector writes the Lifecycle Phase to the Design.PAGE_TWO.Text10 attribute (usually named “Project Name”, Base ID 2016).