




# Agile Product Lifecycle Management

## MCAD Connectors for Agile Engineering Collaboration User Guide

V3.0.3.0

Oracle Part Number - E27203-01

Januar 2012

# Copyrights and Trademarks




**Contact Address (for Support Services see Preface chapter):**

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

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

# CONTENTS

# PREFACE

| | |
|---|---|
| **Note** | To read the PDF files, you must use the free Adobe Acrobat Reader version 7.0 or later. This program can be downloaded from from www.adobe.com. |

## 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.

# Starting the EC Web Connector

Engineering Collaboration (EC) is operated from within your CAD system environment. Your administrator will provide you with a start-up command or icon that will start your CAD system with the Engineering Collaboration functions enabled.

**Note** In order to use Engineering Collaboration you must be a registered Agile user.

Starting with release 3.0, Engineering Collaboration MCAD connectors can work with the PLM Design data model only. This chapter describes the use of EC with the Design data model.

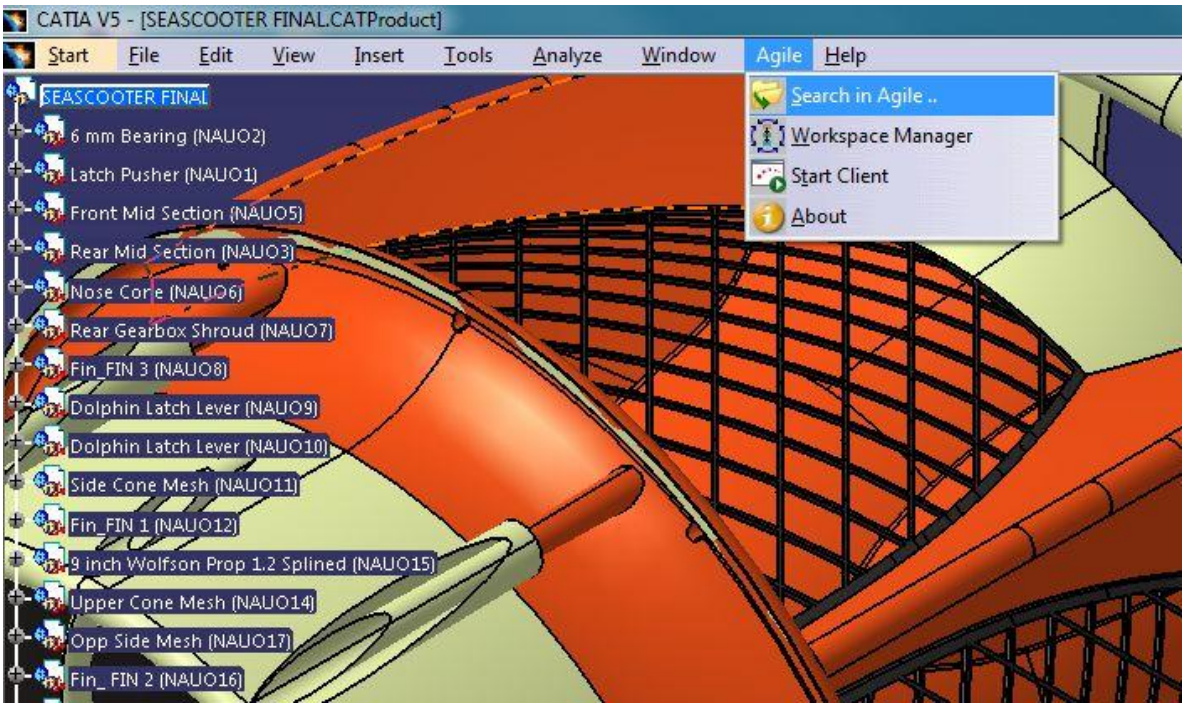
## Menus and Toolbars

When EC is enabled, you will see an Agile menu in your menu bar, and optionally an Agile toolbar. Access to Engineering Collaboration functions is through this menu or toolbar.

**Table: EC Access Methods: According to the CAD System**

| CAD System | EC Access Method |
|---|---|
| Pro/ENGINEER | Menu and toolbar |
| CATIA V5 | Menu and toolbar |
| SolidWorks | Menu |
| Inventor | Ribbon Bar |
| NX | Menu and toolbar |

An example of both the Agile menu and toolbar is seen in the figure below:




The contents and function of the Agile menu and toolbar, which are common to all connectors, are shown in the table below.

**Table: Agile Commands**

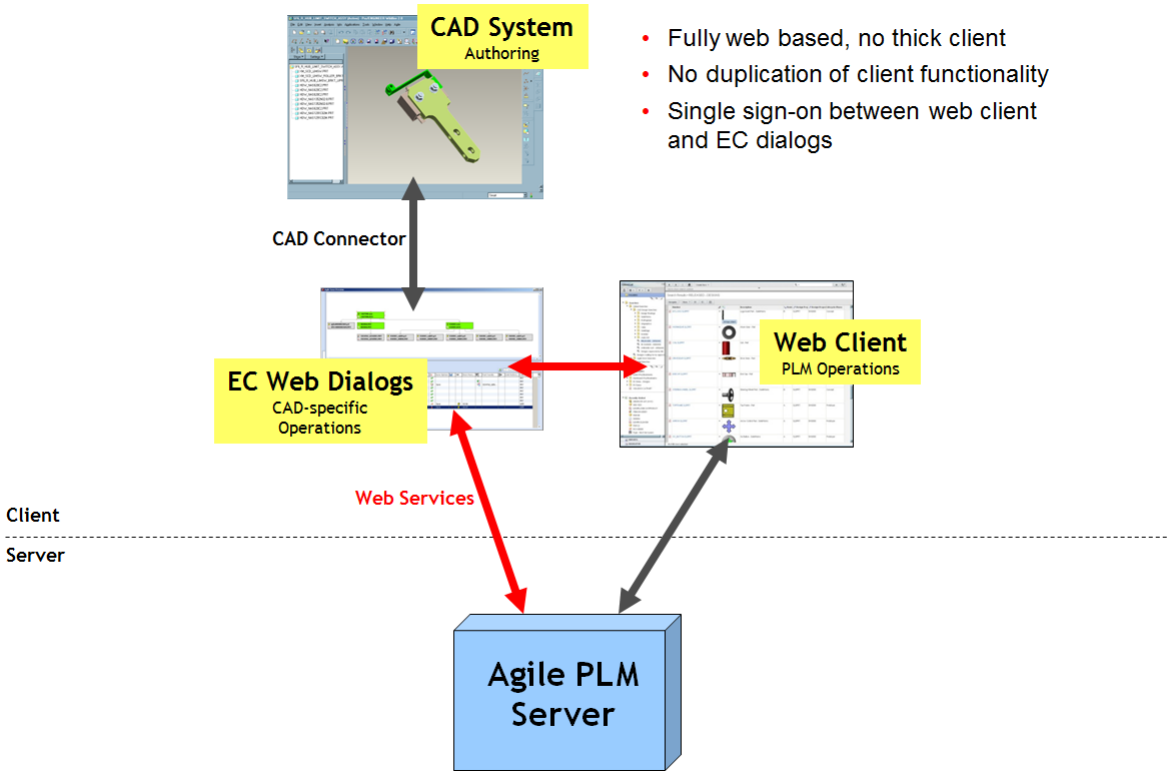
| Command | System | Icon | Function |
|---|---|---|---|
| **Load** | All |  | Opens the Agile Web Client in Parametric Search mode in order to find a design to load into CAD. Note that the load function can be initiated directly from the web client using the *Load to CAD* function. |
| **Workspace Manager** | All |  | Opens the Workspace Browser window. Displays checkout and revision status of the current CAD model and all its components. Also allows changing checkout status,and creation and deleteting of different workspaces. If a Part Family Generic is active in CAD, the status of the entire Part Family table is displayed. |
| **Save** | All |  | Saves files from the current CAD model and all its components into Agile, with a dialog that allows the setting of save options. |
| **Save Session** | All |  | Saves files from the current CAD session into Agile, with a dialog that allows the setting of save options. |

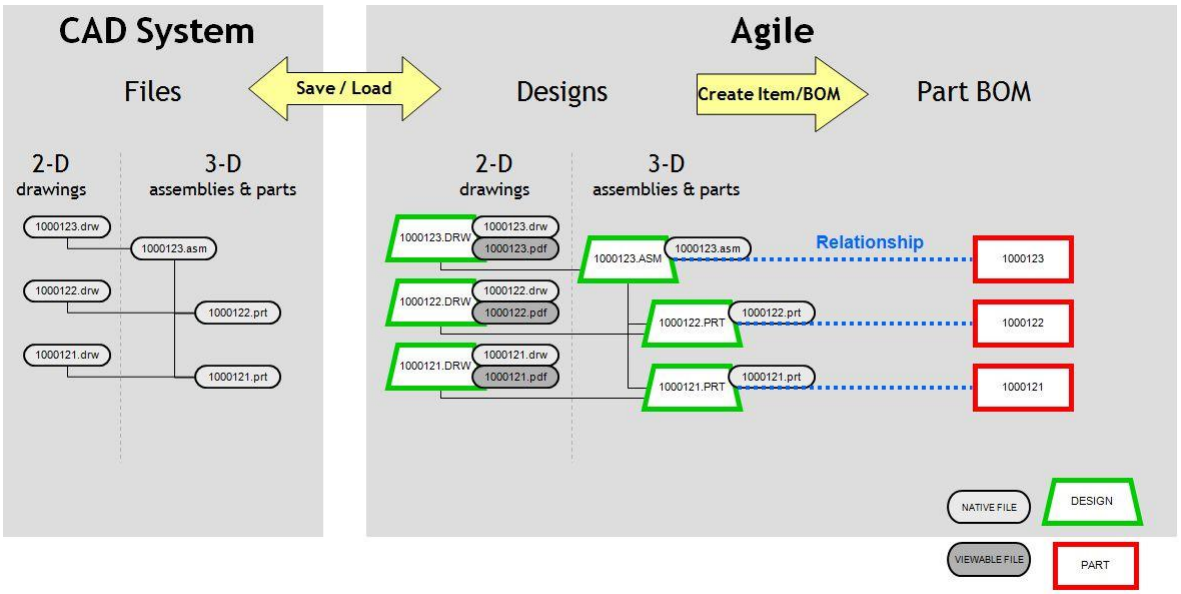
| Command | System | Icon | Function |
|---|---|---|---|
| **Save Family Table** | Pro/E, NX |  | Allows the user to save an entire family table at once. With Pro/E, this process also validates the family table. The user is prompted if there are any errors with the validation process and a log file is created for review. |
| **Show Agile Form** | All |  | Launches the Agile Web Client and displays the Agile form corresponding to the current CAD model. |
| **Update Properties** | All |  | Sets property (attribute) values in CAD based on values from Agile. Properties for the current CAD model and all its components are updated. The specific attributes to map are defined by your administrator in the configuration file. |
| **Update Properties → First Level** | All |  | Same as *Update Properties*, but only sets them for the current CAD model and the next lower level (typically used for setting drawing and model properties together). |
| **Update Properties → Current** | All |  | Same as *Update Properties*, but only sets them for the current CAD model. |
| **Update Title Block** | All |  | Sets text values in the CAD drawing title block based on values from Agile. The specific attributes to map are defined by your administrator in the configuration file. |
| **Insert CGR** | CATIA V5 |  | Adds a CGR file to the current CATProduct. |
| **Save with CGR** | CATIA V5 |  | Saves the CGR file along with the native CATPart or CATProduct file and establishes the relationship between them. |
| **Open Native File** | CATIA V5 |  | Opens the native CATPart or CATProduct files for the selected CGR file(s). |
| **Reload CGR** | CATIA V5 |  | Updates the selected CGR file(s) with the latest version from PLM. |
| **Disconnect Session** | All |  | Used to start (or re-start) the EC Web Connector. Usually the Web Connector is started automatically on demand. |
| **About** | All |  | Displays information about the current version of the CAD Connector. |

# CAD Connector Functionality

In order to understand the details of the CAD Connector functionality, it is important to understand the overall process and how the data is stored in Agile. The figure below gives a high-level view of the process.

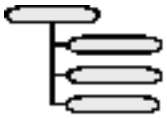
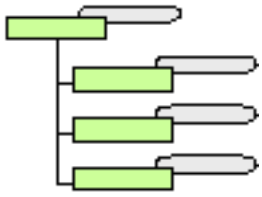
**Figure: Agile Engineering Collaboration Process**

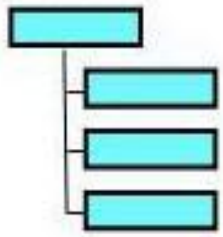





One important concept is that EC creates two distinct structures inside Agile, one is the Design Structure, and the other is the Part BOM. Each of these is based on the CAD file structure that is created by the designer in the CAD system. The following table explains the purpose of each of these structures:

**Table: EC Structure types**

| Structure Type | Data Type | How Created | Purpose |
|---|---|---|---|
| CAD File Structure  | Files | Created by building CAD models. The structure is known within the CAD files. | Defines the assembly structure of CAD models. Also used to structure the relationships between CAD drawings and the components on the drawing. |
| Design Structure  | Agile Design object, with attached files | Created using the EC *Agile → Save* command | Manages the CAD files within Agile, for saving and loading designs. The Design Structure matches the CAD file structure on a one-for-one basis. |

| Structure Type | Data Type | How Created | Purpose |
|---|---|---|---|
| Part BOM  | Agile Part object | Created using the EC *Agile → Save* command with publish option | The Part BOM represents the physical product that you are going to build. When the Part BOM is created by EC, the Design objects can be linked in a variety of ways to the Part BOM. |

To enable you to create and modify these structures in Agile, the CAD Connectors have three main functions, each of which has its own dialog window in the EC Web Connector. These functions are *Save*, *Load*, and *Workspace Manager*. These functions are described in the following sections, followed by other miscellaneous functions.

# Saving to Agile

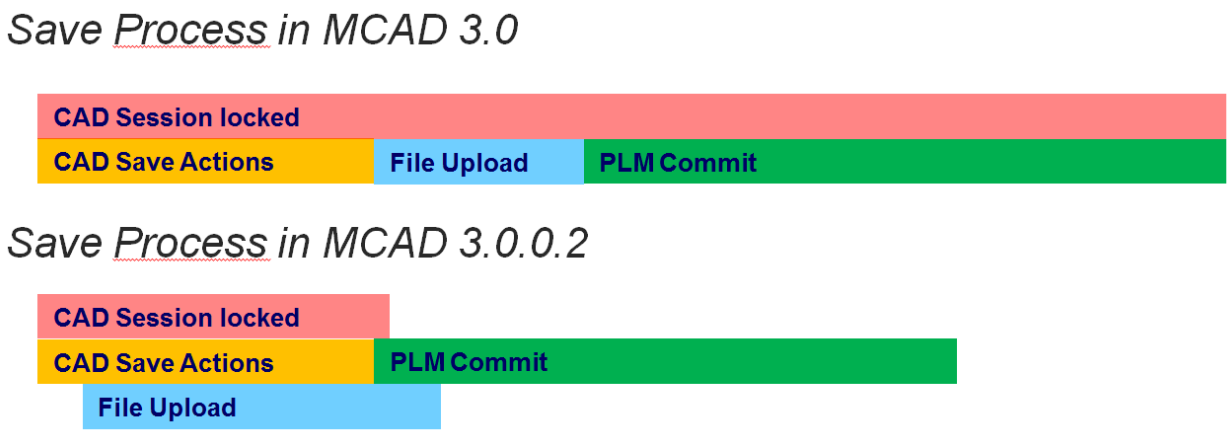
## Introduction

Saving into Agile, using the *Agile → Save* command, creates a Design Structure in Agile. This structure stores all CAD design files (Parts, Assemblies, Drawings, etc.) in a way that supports CAD work-in-progress design, and makes the data available to the rest of the organization, as privileges permit. It will save the current CAD model (whatever is in the active CAD window), including all lower-level components.

## Asynchronous PLM Commit

In order to unlock the CAD session during save an asynchronous file upload and PLM commit is implemented. The switch **handling** set to **asynchron** in the **CAXConfig.xml**

(**`ConnectionProperties`** section) enables this option.



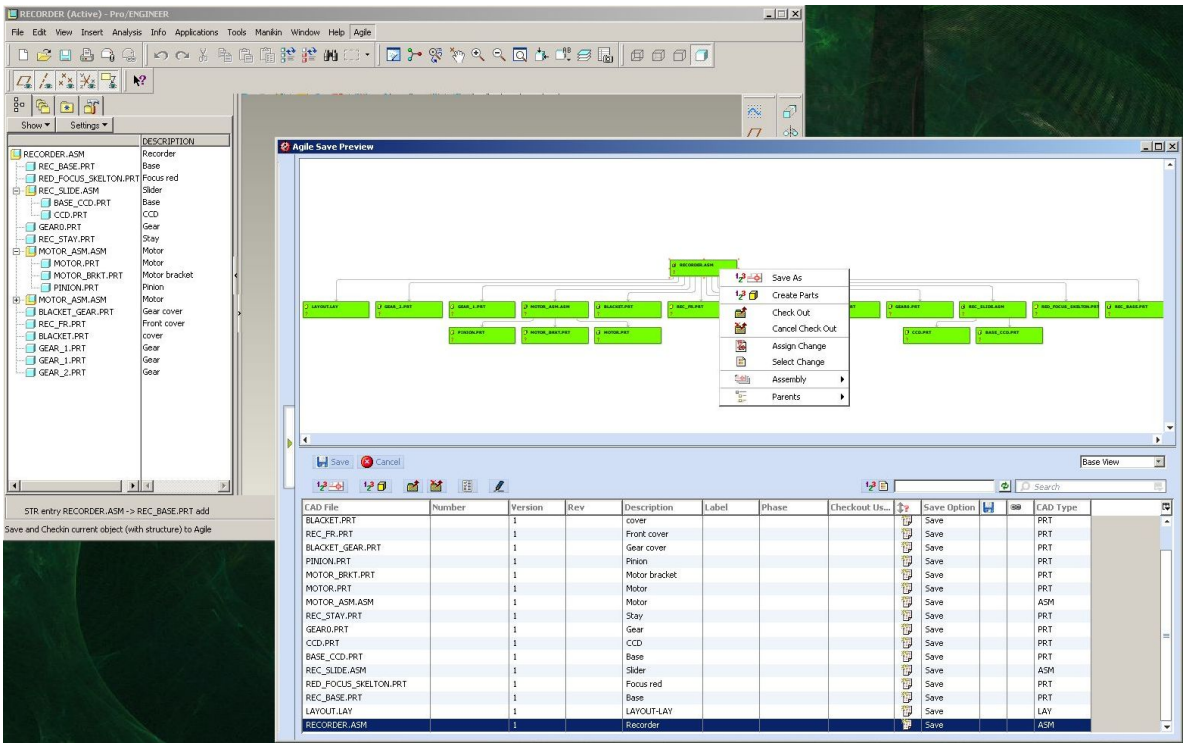

The figure displays the main steps during the save process and shows the differences. The PLM actions for file upload and committing the PLM changes are run concurrently.

The CAD session is unlocked as soon as all necessary CAD actions are done. This reduces the time during save where the CAD system is locked to the minimum. The CAD engineer can continue to work in CAD while the PLM connector is still uploading files and committing the PLM changes. A summary window is displayed after the upload and commit is complete in PLM.

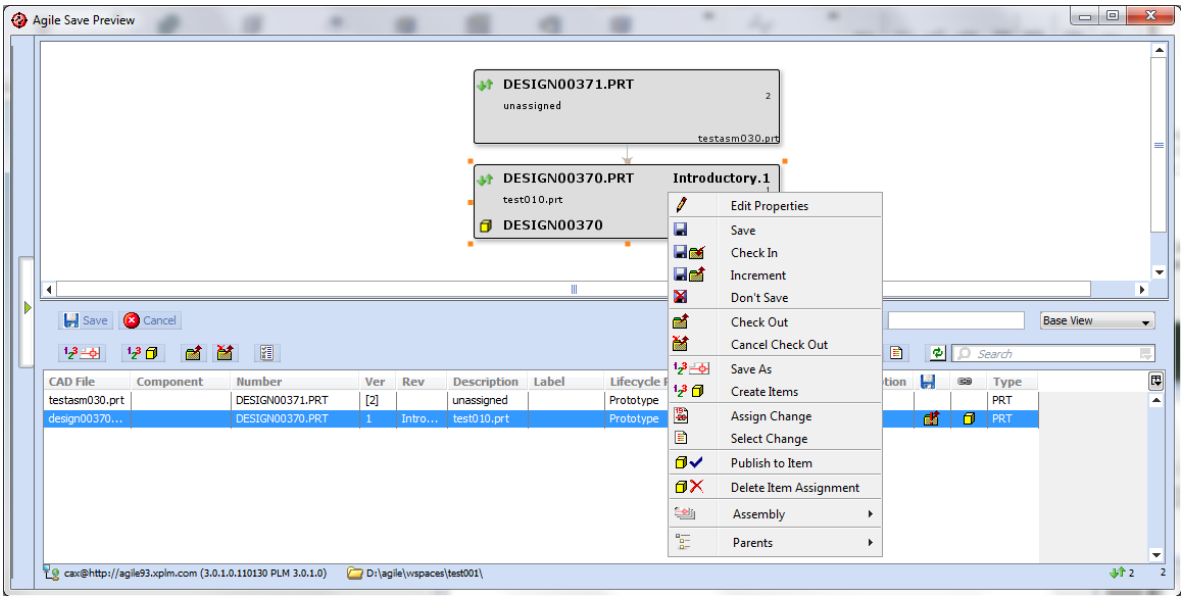
The total time where everything is finished and saved in PLM depends on the network and PLM performance itself. In general the total save time is less than in 3.0.

## Using the Save Command

When you execute the *Agile → Save* command, the Save dialog is displayed similar to what is shown in the figure below.

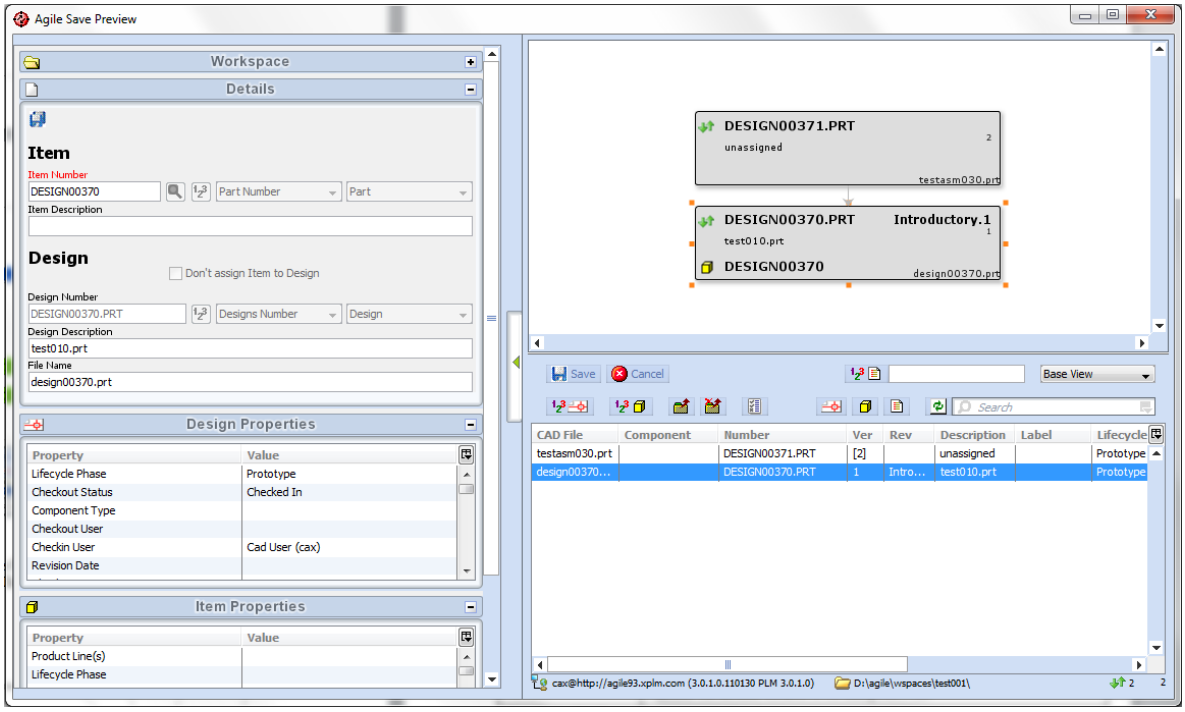


The Save dialog has two display areas. The top area displays the CAD structure information. Each node in this browser view contains the CAD filename, the assigned PLM Design number and PLM related information.




The bottom area displays the CAD attributes and some PLM attributes. The context menu is available in the list view and in the browser view. It contains the same menu items in both views to control the save behaviour.

After changing any options within the dialog (see below for details), you click the *Save* button to start the save process. When saving for the first time, autonumbers will be assigned to each model depending on the preferences default settings. Interactive mode can be used by double-clicking on a node or row in the dialog, which pops up a details dialog out of the left sidebar for the selected file, in order to capture property information needed for the initial save, as shown in the image below.




Details of the Save dialog are shown in the figures *Save Dialog Toolbars* and *Details of Save Dialog* below, and described in the tables *Toolbar Options* and *List Fields and Control*.

**Figure: Save Dialog Toolbars**

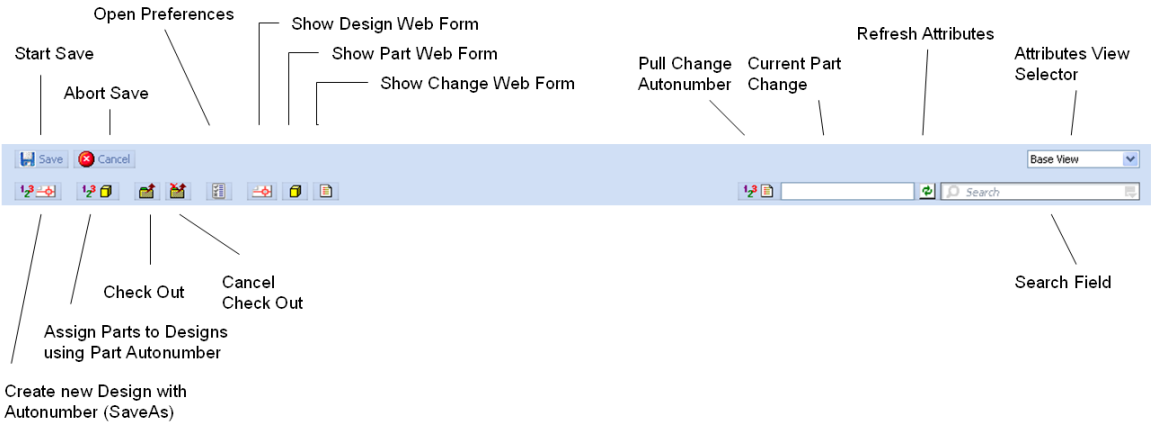


**Table: Toolbar Options**

| | |
|---|---|
| **Start Save** | Starts the CAD Save procedure including upload of files and structure update in PLM.<br>Related Drawings or Manufacturing objects get their number from the underlying 3D model in the case that there is only one model directly referenced. If there is no 1:1 relationship, a design autonumber will be used. |
| **Abort Save** | Closes the dialog and returns to CAD. |
| **Create new Design with Autonumber (SaveAs)** | According to the preferences settings for creation of designs and parts, there are two options:<br>1. The *Save As* preference is set to *Part and Design*: In this case the default Part autonumber source is used to assign new numbers to the selected designs. The system will create a design of the default Design subclass using this number plus the file extension. Additionally the system will create and link a Part of the default Part subclass with the same number, if the *Part Assignment* preference is set to *Create and Link*.<br>2. The *Save As* preference is set to *Design only*. In this case the default Design autonumber source is used to assign new numbers to the selected designs, the default Design subclass will be used. The system will not create or link Parts.<br>Related Drawings or Manufacturing objects get their number from the underlying 3D model in the case that there is only one model directly referenced. If there is no 1:1 relationship, a design autonumber will be used. |
| **Assign Parts to Designs using Part Autonumber** | The system will use the default Part autonumber to assign Parts to Designs that are not already linked to any Part. The system will not assign Parts to Drawings, Helper Parts or Manufacturing objects. Only the 3D model will be linked to a Part. |
| **Check Out** | Set checkout reservation for the selected components. |
| **Cancel Check Out** | Cancel checkout reservation for the selected components. |
| **Open Preferences** | Expands a dialog in the left sidebar which allows you to set the default Design sub-class and number, desired viewable file formats, and default property mappings. See the *Preferences Settings* section for details. |
| **Refresh** | Updates the attribute contents of the dialog from PLM. |
| **Show Design Web Form** | Launches the PLM Web Client Form for the selected Design. |
| **Show Part Web Form** | If a Part is linked to the selected Design, the Part form will be launched in the Web Client. |
| **Show Change Web Form** | If a Part is linked to the selected Design and the Part has been assigned to a Change object, the Change form will be launched in the Web Client. |
| **Pull Change Autonumber** | This button pulls a new number from the server using the default Change autonumber source. The number will be written to the Current Part Change field. This Change can then be assigned to Parts using the *Assign Change* context menu command. |

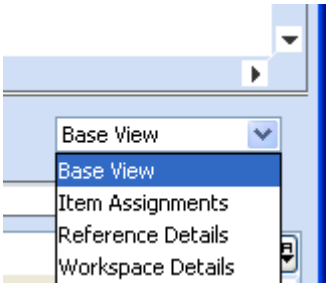
| | |
|---|---|
| **Current Part Change** | This field contains the current Change number. This number can be assigned to selected Parts using the *Assign Change* context menu command or reset using the *Select Change* context menu command. If a part gets attached to this change, the change will be created in the default change subclass. |
| **Attributes View Selector** |  <br>The view selector switches the visibility of attribute sets in the list area. The *Base View* contains the most important information for tracking the PLM checkout status and the assignment of a CAD file to a PLM design object. The *Item Assignments* view shows additional information about the Part assignment and the Part attributes. The *Reference Details* view displays specific information for parts with external references or part families. The *Workspace Details* view shows the complete local path settings in addition to the *Base View*. |

**Figure: Details of Save Dialog**

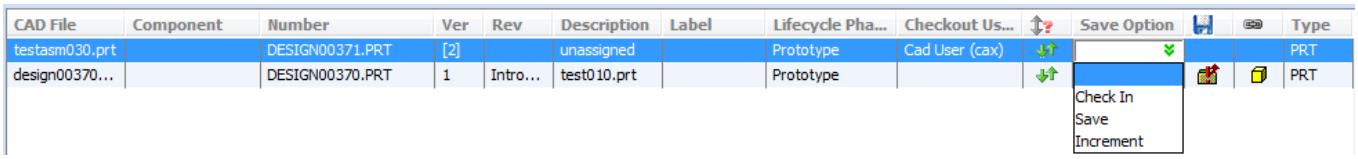
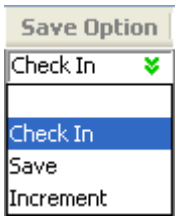



**Table: List Fields and Controls of the Base View**

| | |
|---|---|
| **CAD Object** | The CAD filename that is being saved. |
| **Number** | Number of the Design objects in Agile that contain the CAD file and any viewable files. |
| **Component** | Component type of the Design object in Agile. |
| **Version** | Version of the Design object. This number starts at 1 and increments by 1 for each check in. |
| **Rev** | The Revision of the Design object, which includes a major and minor component. Each version has a unique revision, which is controlled by the part revision logic. |
| **Description** | Description of the Design object in Agile. |
| **Label** | Label for the specific version of the Design object. |
| **Lifecycle Phase** | The lifecycle phase associated with the current Design version. |
| **Checkout User** | The name of the current checkout user, if any. |

| | |
|---|---|
| **File Status**  | Indicates whether the file has been modified in CAD. If so, the Modified Flag icon will be displayed in the column, and the *Save Mode* will be set to save the file.<br>There are the following statuses:<br>  – The object is new to PLM, the object will be preselected for save<br>  – The object is up to date with PLM, the object will be deselected from save<br>  – The object is known in PLM and is modified locally, the object will be preselected for save<br>  – The object is known in PLM and was modified by someone else in PLM, the object will be deselected from save<br>  – The object was modified in PLM and locally. The object will NOT be preselected for save. The user decides whether to overwrite the changes in PLM or not. |
| **Save Mode**<br>  | Controls whether the file is saved or not, and in what mode. There are four possible options, that can be selected directly in the user interface:<br>▫ **(blank)** – don't save<br>▫ **Check In** – save and check in (release reservation)<br>▫ **Save** – save and keep checked out (keep reservation)<br>▫ **Increment** – save, check in, and check out again<br>**Note:** Check in is the only option that will trigger publishing based on the preferences settings. |
| **Save Status**<br>  | Indicates whether or not you have the ability to save this component into Agile, based on your privileges and the state of the object in Agile. If not, a *Stop Sign* icon will be displayed in the column. If there is an action required, an exclamation mark is shown.<br>Indicates the progress of the save operation as follows:<br>  – File has been successfully saved<br>  – No write privilege in PLM, save process is stopped at this point<br>  – Checkout or SaveAs Action required, in order to save the file to PLM<br> – In PLM a newer version is available. Checkout only if you want to overwrite the PLM version with the local version. You need to select the save option manually even if the object is marked as local modified. If you don't want to overwrite, use the Save As command in order to assign a new Design object.<br>  – This sign explains the missing privileges in combination with the underlying icon. The privileges for checkout (  ), check in (  ) and modify (  ) are checked. If another user has checked out the object, the cancel checkout is disabled (  ).<br>You can only use the *Save As* command to create a new Design object in PLM. You cannot overwrite the existing Design in PLM. |

| | |
|---|---|
| | **(blank)** – The file can be saved to PLM. |
| **Part Assignment**  | Indicates whether or not an Item is assigned to the Design<br>  – Design will not get an Item assigned (Drawing, Helperpart, Manufacturing)<br>  – Design is assigned to an Item object<br>**(blank)** – Design is not assigned to an Item object |
| **Item** | This field indicates the Part Number:<br>▫ When the Item is initially being created, this displays the pre-defined mapping of the Item Number field for this item. This can be overridden in the interactive dialog.<br>▫ Once the Item has been created, this field will show the Item Number that was used. |
| **Rev** | Current revision of the Item in Agile. Parentheses indicate a pending revision. |
| **Change** | This shows the ECO number that is assigned to this Item, to control the Item creation or update through the change process. If there are multiple pending changes, the desired change can be selected here. |
| **Description** | Description of the Item object in Agile. |
| **Lifecycle Phase** | The lifecycle phase associated with the current Item. |
| **CAD Type** | Shows the CAD file extension, which can be used for sorting. |

Details of the interactive Save Dialog are shown in the figure below and described in the following table.

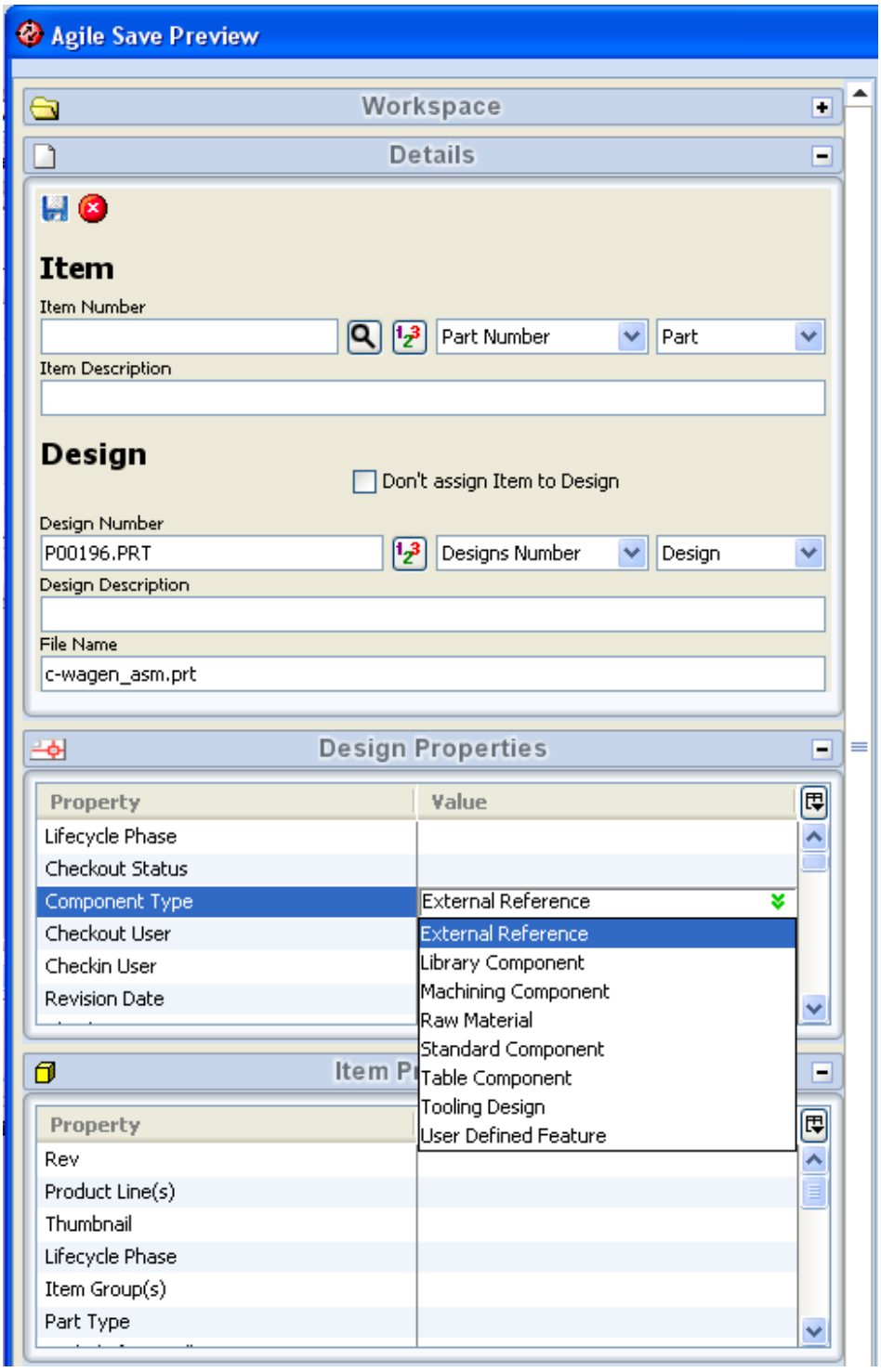
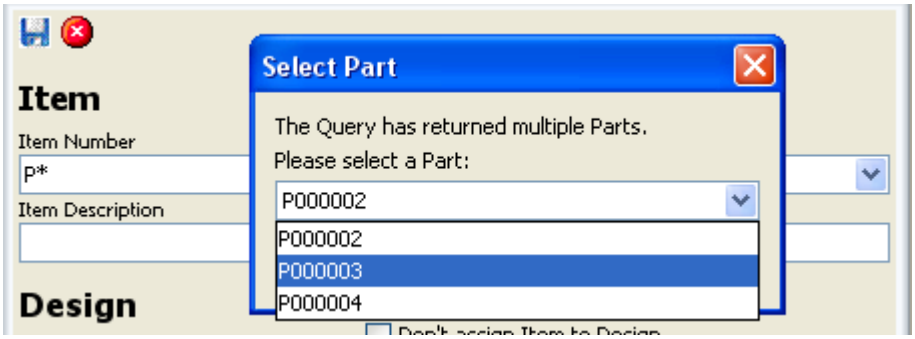


**Table: Details of the Interactive Save Dialog**

| | |
|---|---|
| **Item Number** | The Item Number that is or will be assigned to the Part. If no Design number is set, this value will also be written to the Design Number field with the CAD file extension appended. |
| **Item Description** | The description of the Part item that is or will be assigned to the Design. |
| **Item Search**  | This button executes a Part Search in PLM using the values entered in the Item Number and Item Description field. If there is one match, the found number is written into the Part Number field. If more than one match is found, a dialog pops up and the user can select the desired part from the list of found parts. This is used to assign the Design to an existing Part.   |
| **Item Autonumber Button**  | If you click this button, it will put the next available auto number from the selected Part sub-class and Part auto number into the Item Number field. |
| **Part sub-class** | Selector which allows you to pick the Part sub-class to use for saving this particular CAD file into Agile. This is for overriding the default value set in the main Preferences dialog. |
| **Part Autonumber** | Selector for the Part auto number to use for saving this particular CAD file into Agile. This is for overriding the default value set in the main Preferences dialog. |
| **Design Number** | The value that will become the number assigned to the Agile Design that is being created. |
| **Design Description** | The value that will become the description assigned to the Agile Design that is being created. |
| **File Name** | The CAD file name that is being saved into Agile. |
| **Design Autonumber button**  | If you click this button, it will put the next available auto number from the selected Design sub-class and Design auto number, into the Design Number field. |
| **Design sub-class** | Selector which allows you to pick the Design sub-class to use for saving this particular CAD file into Agile. This overrides the default value set in the main Save dialog. |
| **Design Autonumber** | Selector for the auto number to use for saving this particular CAD file into Agile. This overrides the default value set in the main Save dialog. |

| | |
|---|---|
| **Property / Value area** | This area displays, and allows editing, for other properties that are being set from CAD into Agile. Values may be pre-populated based on the settings in the Save Preferences dialog. |

**Note** Your site most likely has pre-defined mappings for Number, Description, and other properties. You should check with your administrator to understand the allowable values to use. Also, these properties can be set as *Required*, meaning that you must enter a value before exiting the dialog.

## Multi-Select and Context Menus

Since you will often have many items listed in the Save dialog, it is convenient to be able to set options for multiple items at a time. This is made possible by multi-select and context menus. The context menus are available in the browser view and in the list table as well.

The browser view provides functionality to select dependent assembly trees or parents beside the common multi-select functionality. The *Assembly* sub menu executes the operations for all children of a selected node recursively. The *Parents* sub menu executes the operations for all parents of the selected node recursively.

To multi-select, simply click within any item in the browser or list view, and either hold down on the left mouse button and drag the cursor, or use Shift-click or Control-click. Once you have selected the desired items, you can use the context menu (right mouse button) to execute any of the commands listed in the following table.

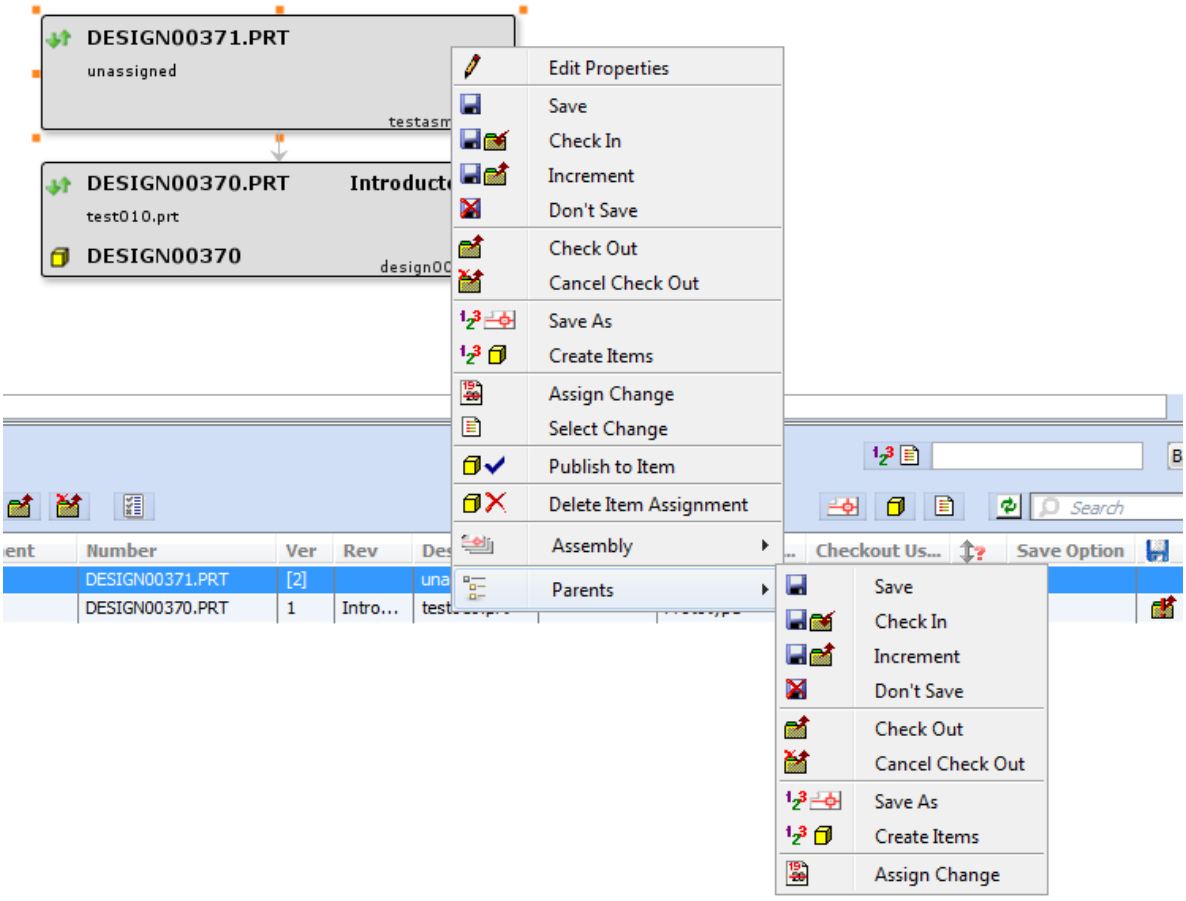



**Table: Save Dialog - Context Menu**

| Command | Description |
|---|---|
| **Edit Properties** | Edit Metadata of multiple selected objects in the sidebar. Multiple values are displayed as "***". If all selected objects have the same value, the value is displayed. If the user wants to keep the values with no changes to the selected objects, the "***" can be used as a value in multiedit mode. |
| **Save** | Sets all selected objects to be saved, overwriting the file in the current version. |
| **Check In** | Sets all selected objects to be saved and checked in. |
| **Increment** | Sets all selected objects to be saved, checked in, and checked out again. |
| **Don't Save** | Sets all selected objects to NOT be saved. |
| **Check Out** | Sets checkout reservation for currently selected objects. |

| | |
|---|---|
| **Cancel Check Out** | Cancels checkout reservation for currently selected objects. |
| **Save As** | According to the preferences settings for creation of designs and parts there are two options:<br>1. The *Save As* preference is set to *Part and Design*: The default autonumber source is used to assign new numbers to the selected designs. The system creates a design of the default Design subclass using this number plus the file extension. Additionally the system will create and link a Part of the default Part subclass with a new number, if the Part Assignment preference is set to *Create and Link*. If the Item and Design numbering is set up to be equal, then the Design number equals the Part number, plus the file extension.<br>2. The *Save As* preference is set to *Design only*: In this case, the default Design autonumber source is used to assign new numbers to the selected designs. The default Design subclass will be used. The system will not create or link Parts.<br>Related Drawings or Manufacturing objects get their number from the underlying 3D model in the case that there is only one model directly referenced. If there is no 1:1 relationship, a Design autonumber is used. |
| **Create Parts** | The system will use the default Part autonumber to assign Parts to Designs which are not already linked to any Part. The system will not assign Parts to Drawings, Helper Parts or Manufacturing objects. Only the 3D model will be linked to a Part. |
| **Assign Change** | Assigns selected items to the Change shown in the *Current Change* field in the toolbar at the top of the list. |
| **Select Change** | If a Part with a Change is assigned to the selected Design, the Change Number is written into the Current Change field in the toolbar at the top of the list. This is to allow this same Change to be used for additional objects, using the Assign Change function above. |
| **Publish to Item** | Publish the latest checked in Design content and structure to the assigned Item. |
| **Delete Item Assignment** | Remove the relationship to the Item in PLM. |
| **Assembly Sub Menu** | Executes all contained actions for all children of the selected nodes. |
| **Parents Sub Menu** | Executes all contained actions for all parents of the selected nodes. |

# Numbering Options

There are several numbering options that depend on the preference settings explained in the following chapters. Possible options for numbering of the filename, the Design number and the Part numbers are:

| Filename | Design Number | Part Number | Creation Sequence |
|---|---|---|---|
| Don't Rename | Filename | Part Autonumber | Design and Part together |
| Rename | Design Autonumber + Ext | CAD Property | Design only |
| | Part Number + Ext | Existing (select) | |
| | CAD Property + Ext | | |
| | Drawing Logic (get Model Number) | | |
| | Helper Part Logic (get always a Design number) | | |

The valid combinations according to the preference settings and CAD Model types are:

| CAD Type | Filename | Design Number | Part Number | CreationSequence |
|---|---|---|---|---|
| Model | Don't Rename | Filename | (any) | (any) |
| Model | (any) | Part Number + Ext | (any) | Drawing and Part together |
| Model | (any) | Design Autonumber + Ext | (any) | (any) |
| Model | (any) | CAD Property + Ext | (any) | (any) |
| Drawing | (any) | Drawing logic | NONE | -- |
| HELPERPART | (any) | Helper Part logic | NONE | -- |
| Format | (any) | Helper Part logic | NONE | -- |
| Manufacturing | (any) | Helper Part logic | NONE | -- |
| Part Family | (any) | Helper Part logic | NONE | -- |

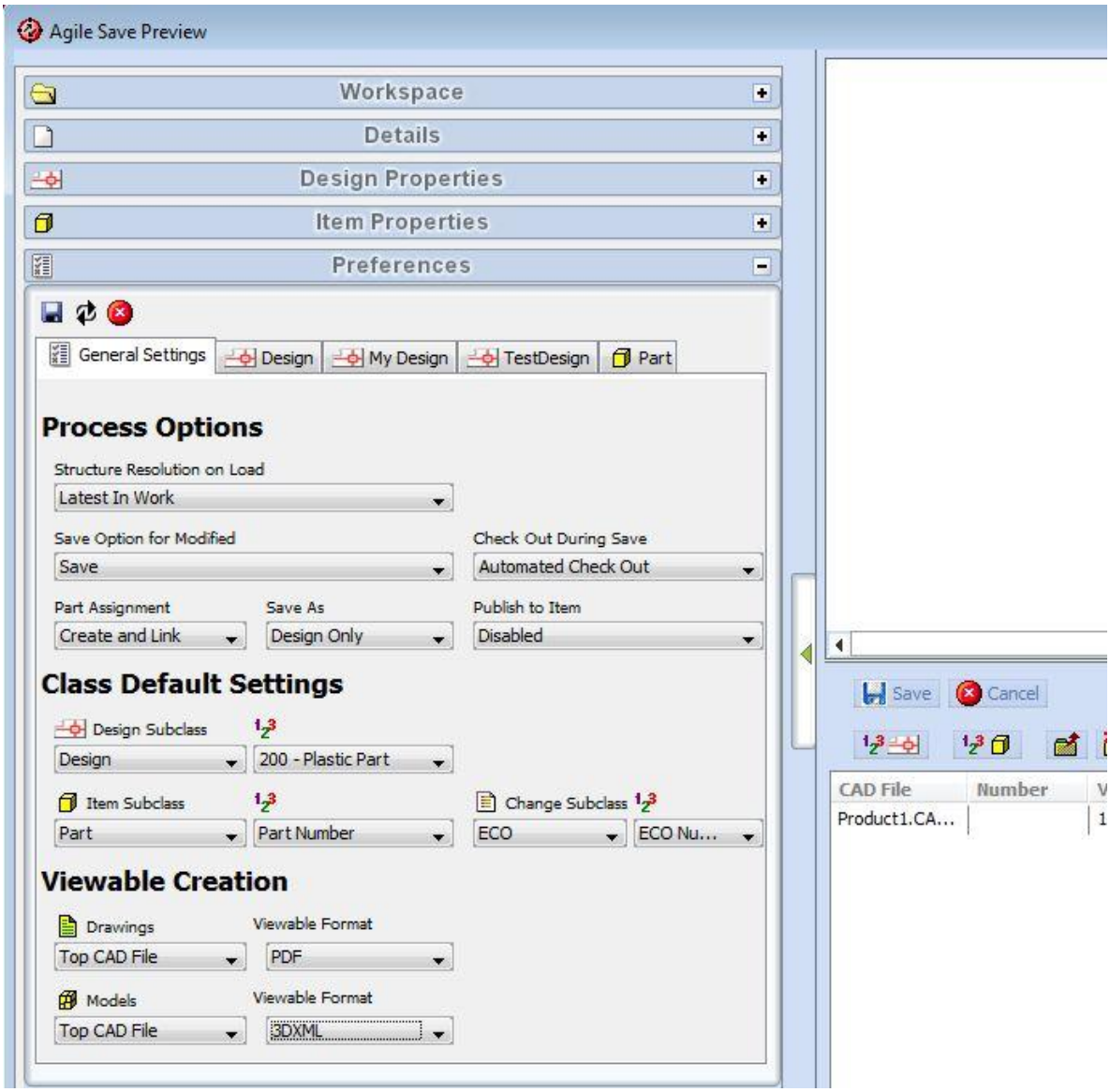
For Drawings and Manufacturing objects the number is generated from the child model if there is only one model assigned. In this case the model number is used for the drawing. If there is already an existing drawing with the same number in the system, the logic appends a counter to the drawing number in order to not overwrite existing drawings or Manufacturing objects. No Part is created, because it is already created by the child model.

If there is more than one model referred from the Drawing or Manufacturing object, a Design autonumber is used and no Part is created.

For Formats and Helperparts, the system will not create any Part. Only a Design autonumber is used regardless of the *SaveAs* preference setting.

# Preferences Dialog

The Preferences dialog is accessed using the button in the Save or Load dialogs or by expanding the left sidebar and the contained preferences container, in addition it can be accessed from the Create Object dialog via expanding the preferences container.

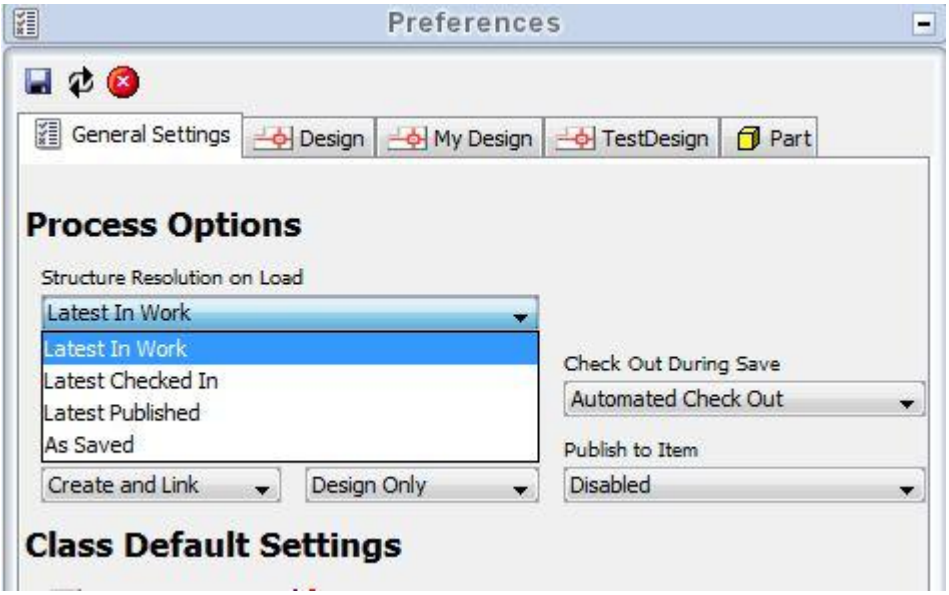



The *Process Options* define the behavior during load and save operations. The *Class Default Settings* predefine the default subclasses and autonumbers to be used, if new Parts, Designs or Change Orders are created. However, if the Create Object dialog is used for object creation, the default settings concerning subclasses and autonumbers, because the predefined subclass of the

template choosen in that dialog is preffered over this one.

# Load Preferences

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

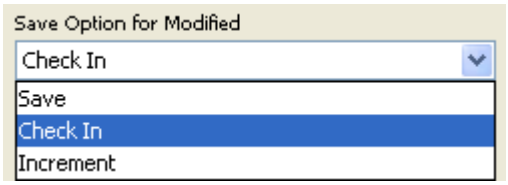



The valid values and their meaning are:

| | |
|---|---|
| Latest in Work | Select the latest possible Design version of a component, including versions that are currently checked out by the current user. |
| Latest Checked In | Select the latest Checked In Design version of a component. |
| Latest Published | Select the latest Design version, which is a Part attachment. |
| As Saved | Select the Design version that was saved within the parent assembly. |

# Save Preferences

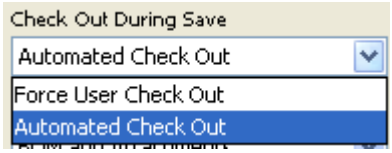
**Default Save Option**




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

| | |
|---|---|
| Save | The file is saved into the currently checked out Design version. The Design remains checked out after save. |
| Check In | The file is saved into the currently checked out Design version and then the Design is checked in. |
| Increment | The file is saved into the currently checked out Design version. Then the Design is checked in and then checked out again immediately. The Design remains checked out after save with an incremented version. |

**Checkout during Save**



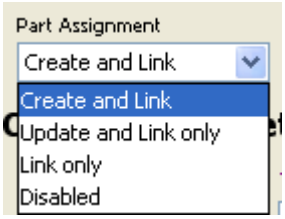

The checkout behaviour during save is controlled by this switch. The valid options are:

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

# Item and Publish Preferences

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

**Part Assignment**




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

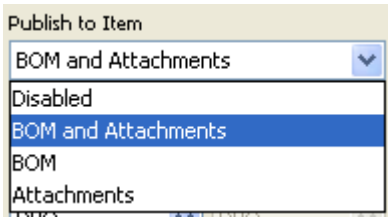
| | |
|---|---|
| Create and Link | This creates new Part objects, if a new Design is created. The Part is linked to the Design and the Part properties are also updated. |
| Update and Link only | This option does not create Parts. Existing Parts are linked to the Design and the Part properties are updated. |
| Link only | This option 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. |

**Save As Behaviour**


The *Save As* option controls whether or not Parts will be created during Save As of a CAD model. The valid options and their meanings are:

| | |
|---|---|
| Design Only | Design Autonumber is used for new Design objects. No Part objects are created. |
| Part and Design | Part 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”. The Part objects get the same number as the Design objects. |

**Publish Behaviour**




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

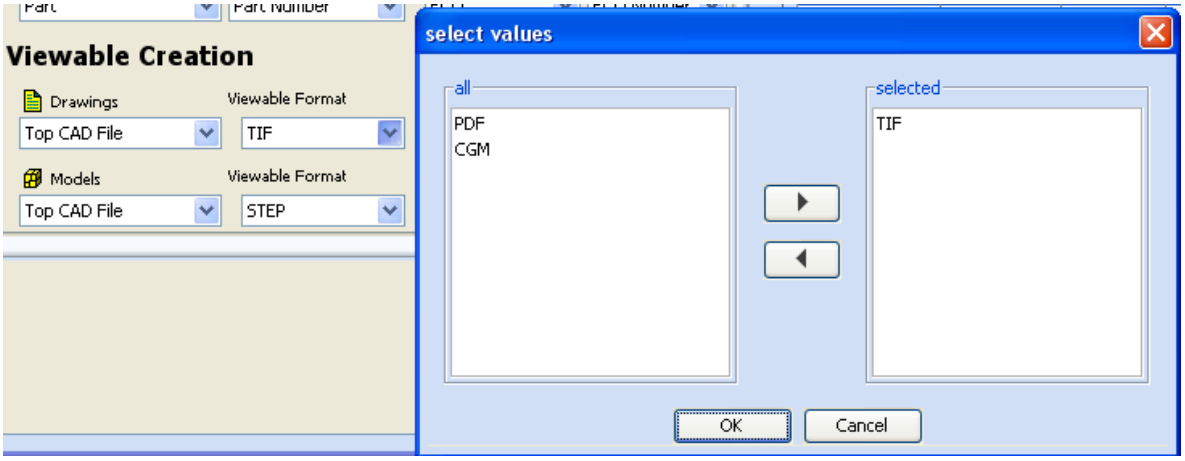
| | |
|---|---|
| Disabled | No Part BOM is updated and no attachments are updated. |
| BOM and Attachments | Part BOM is updated and the Design files are attached to the Parts. |
| BOM | Part BOM is updated. No Design files are attached. |
| Attachments | No Part BOM is updated. Design files are attached.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

# Class Preferences


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

# Viewable Creation Preferences

.




**Viewable File Creation Preferences**: 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 (please contact your administrator). The available Viewable Types are defined in the **CAXConfig.xml** in the viewables structure:

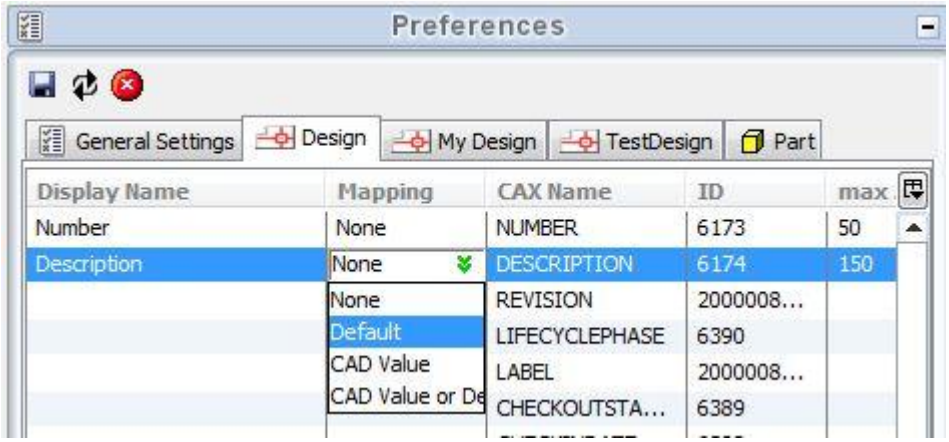
…

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

…

# Property Value Preferences

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




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

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

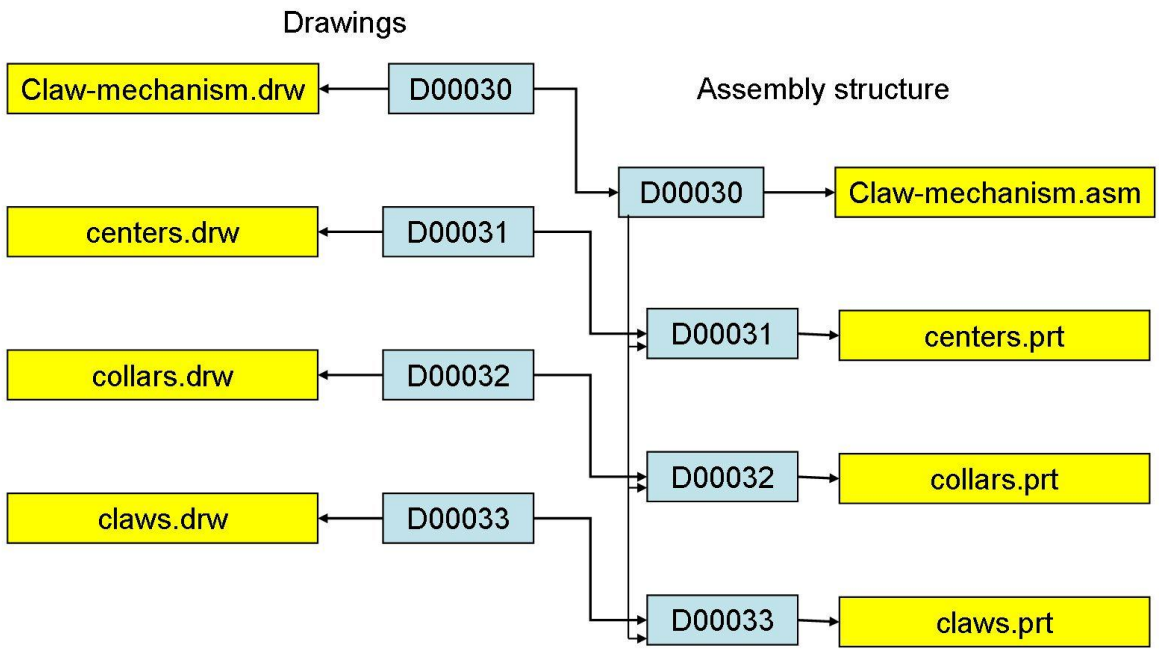
# Design Structures

The *Save* command creates Design structures in Agile to hold the CAD files. It is important to understand how these structures are set up in Agile. CAD designs consist mainly of three types of files – Parts, Assemblies, and Drawings. In terms of the design structure hierarchy, Parts are the lowest level. They are combined into parent Assemblies, which in turn are combined into higher-level Assemblies. Drawings are represented at a higher level than either Parts or Assemblies; essentially a Drawing is a *parent* of whatever Assembly or Part that is on the Drawing.

The figure below illustrates a case of a Pro/Engineer five-part assembly, where each of the parts, and also the assembly, has corresponding drawings. The assembly (D00007) is the parent of all the parts, and each drawing is the parent of its respective assembly or part. It is important to keep this structure in mind when using EC operations like Save and Load, and also when browsing through design structure data in Agile.

For example, since the *Save* and *Load* commands work on a single structure at a time, you can save drawing D00008 and it will save assembly D00007 and the five parts, but not the five-part drawings (because they are not in the same tree structure – you would not see them within the BOM tab of D00008 in Agile). For this situation you can use the *Save Session* command, which will include all files active in the CAD session within the Save dialog.

**Figure: Design Structure Hierarchy**



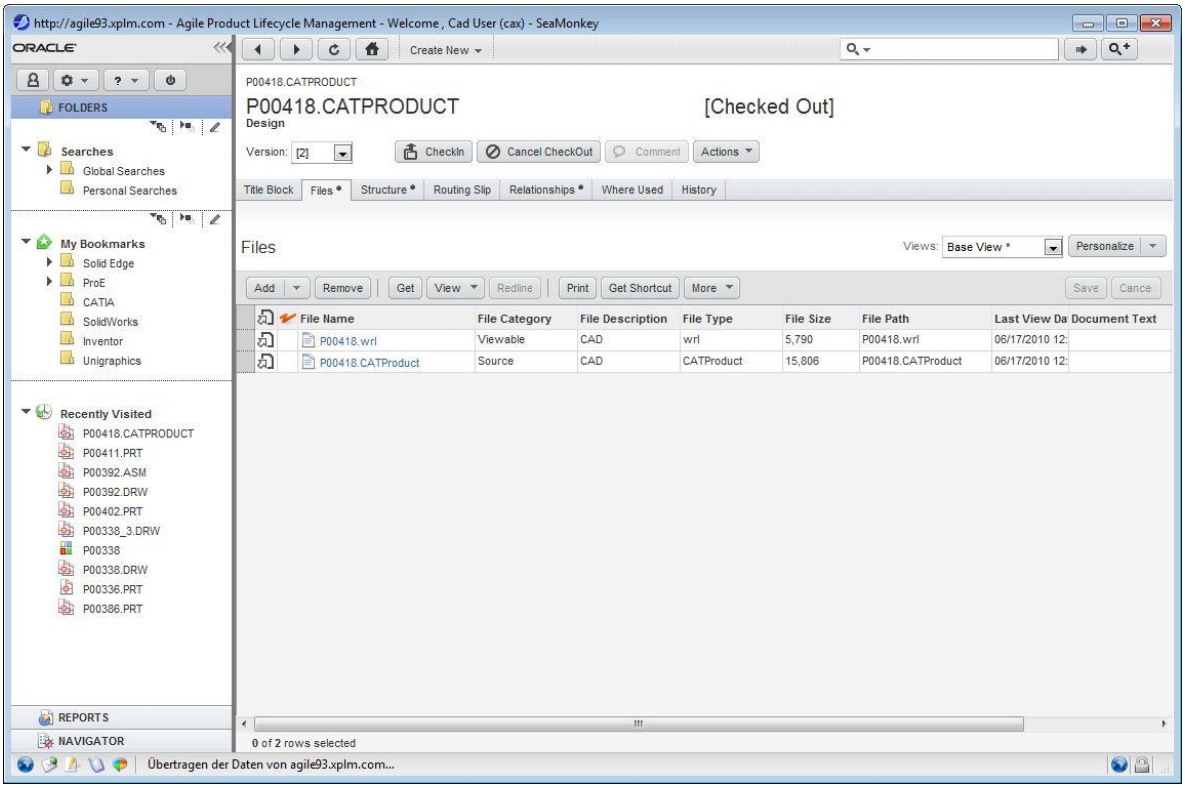
Similar types of structures are used to store other specialty types of CAD files, such as Pro/E family table generics and drawing format files. In each case, the connector is set up to recognize the files as being stored in these particular types of structures, so they must not be manually modified in Agile or errors may result.

# Saving with Derived Files

By configuring the *Save Preferences* dialog (see page 29) you can set the system to save additional derived or viewable files, such as PDF or IGES, in combination with the regular save operation. The reason these operations are combined is to make sure that both the additional file and the native file(s) are in synch with each other. The additional files are attached to the same Agile Design object as the native CAD file. The available file format options within the *Save Preferences* menu are customizable for your site; your administrator will configure this as appropriate.

The figure *Derived File Saved into Agile* shows an example of the results of using this function to generate a WRL file in addition to the native CATIA CAD file.

**Figure: Derived File Saved into Agile**



| | |
|---|---|
| **Note** | In addition to this function with the Agile menu, it is also possible to manually attach files to Agile Designs using the *Add* button on the Design's form in the Agile Web Client. |

# Other Save Commands

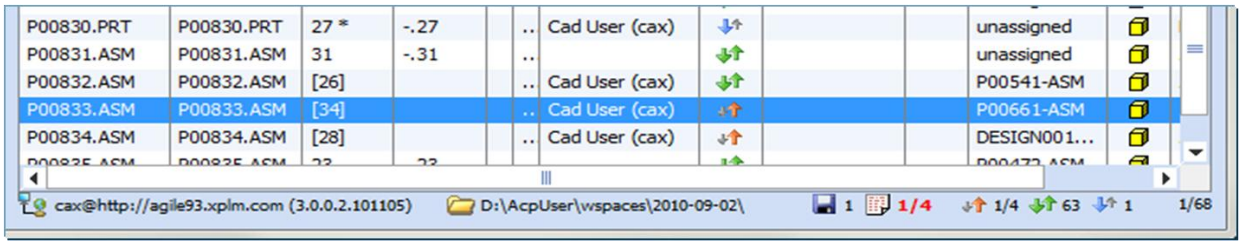
One additional Save command is available:

**Save Session** – This command lists the entire contents of the current CAD session in the Save dialog, rather than just the components within the active model. This is useful especially when you have multiple drawings active in a session, because they can then be saved all at once.

| | |
|---|---|
| **Note** | When you work with Agile Engineering Collaboration you still have a local workspace directory where your CAD files reside. The location of this workspace directory is determined by your system administrator. This is where files are copied to when you use the **Load** command. If you use your CAD system's **File → Save** command, the files will be saved into this workspace directory. You should use **File → Save** to save periodically as you normally would, to prevent data loss during your daily work. Use the **Agile → Save** command on a regular basis to secure your data within Agile, and to make it available to others. |

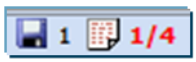
# Status Information

Status information is displayed at the bottom of the dialogs. Beside the current login and server connection, the current workspace and status information is shown.



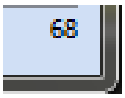

Depending on the statuses and the user selection the status indicator shows the numbers of selected items for save or load. It also shows the number of resolved conflicts per status, highlighted in red bold numbers.

The status indicator is separated into 3 sections. From left to right:




**Main Section** – showing how many files are selected for saving or loading and how many files are modified locally in summary. If the modified file shows up in red bold numbers, not all modified files are selected for saving. The sample shows that one out of 4 modified files is selected for saving.

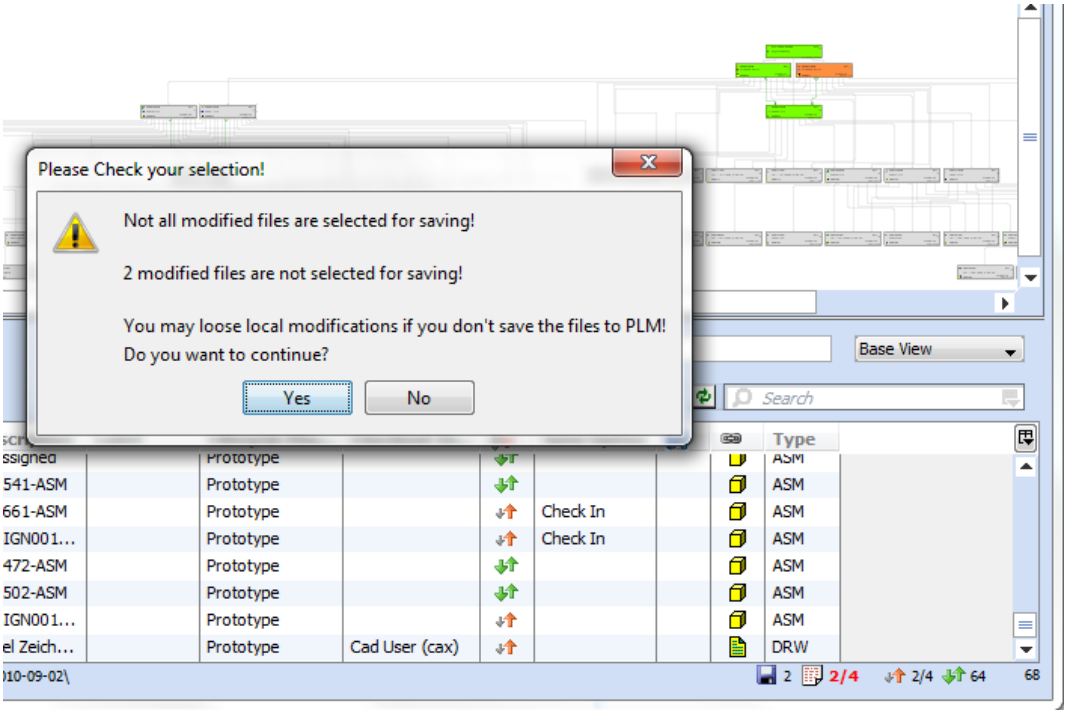
**Details Section** – showing how many files of each status are contained in the list. If the numbers beside the icon show in red bold text, not all modified files of this status type are selected for saving.




**Summary Section** – showing the total object count in the list.

# Guided User Input

Corresponding to the status information additional checks being executed when the user hits the *Save* button in the *Save Preview*. If the user has no modified parts or not all local modified parts selected for saving, a message box is displayed to warn the user. The user can verify the selection again.




There are four possible cases for a message box as shown below:

- Nothing is selected for save – The user must mark something for *Save*, *Check In* or *Increment* in the save option column.

- Local modified file is not selected for save (  )

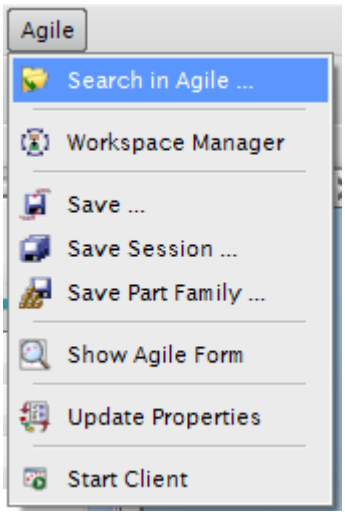
- Local modified file with a PLM conflict is not selected for save (  ) – The user has to decide which objects shall be saved and which objects should not be overwritten in PLM.

- A file with unknown status is not selected for save (  ) – This can come up if CAD mapping defines the Design number (e.g. legacy data) and there is no local cache record available that can detect whether the file is actually in PLM or not. The user has to decide which objects shall be saved and which objects should not be overwritten in PLM.

# Loading from Agile

## Introduction

The loading process from Agile into CAD is executed using the Agile Web Client. The *Load to CAD* action must be enabled in the Agile server configuration. The *Agile → Search in Agile* command can be used to launch the parametric search web form:




If the CAD system is running, however, the Load function **does not need to be** initiated from the Agile menu in CAD. You can simply search within the web client to find the desired Design, and then use the "Load to CAD" function to send the design to CAD.

The Load procedures retrieve previously saved CAD files in order to perform CAD work. Once the desired model is located and selected, the necessary files are extracted from Agile and placed in the designated working directory. The files extracted depend on the type of model selected, as follows:

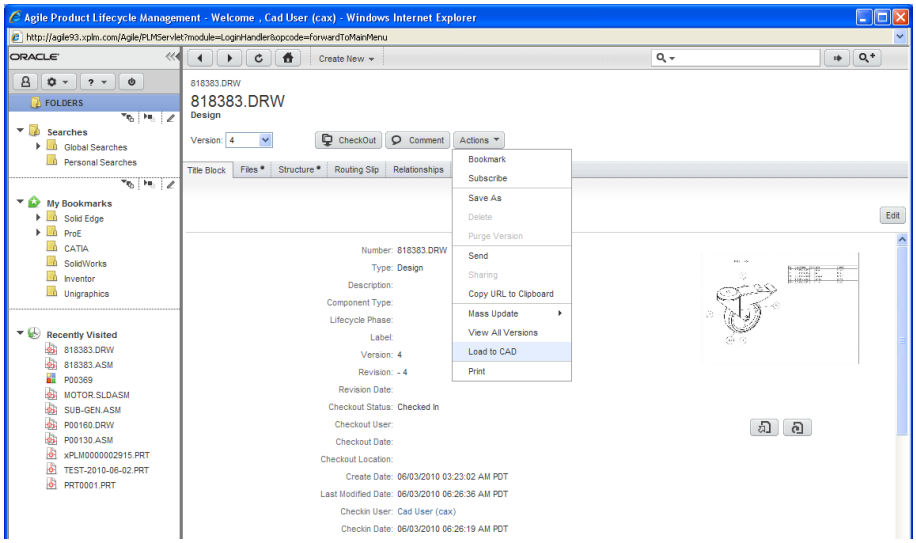
- **Part** – If a single Part is selected, just that single Part file will be loaded to the working directory.
- **Assembly** – If an Assembly is selected, its file and all subordinate files (Subassemblies and Parts) necessary to build the Assembly are extracted.
- **Drawing** – If a drawing is selected to load, its file and all subordinate files necessary to build the Drawing (including all subordinate Assemblies, Subassemblies, and Parts) are extracted.

# Using the Load Command

To load CAD files from Agile into your CAD system, pick the *Load to CAD* command in the Web Client. The *Load to CAD* function is available in the following locations in the Web Client:

- In any search results that include Design objects, in the *More* menu
- From any Design object form, in the *Actions* menu
- From any Item Attachments tab, in the *More* menu

**Figure: Load to CAD Action**

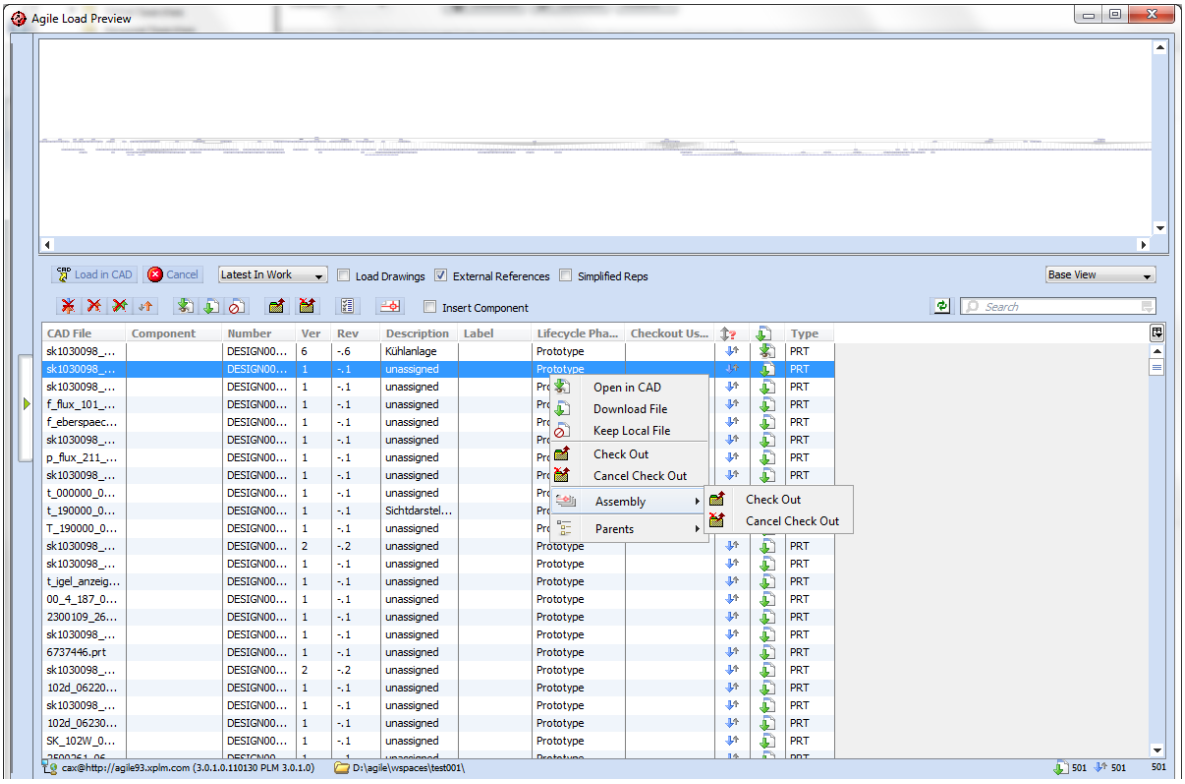



This will bring the EC Web Connector Load Preview forward. If you don't see this window, you will not be able to load.

# Load Dialog

The main purpose of this dialog is so that you can review which files and which versions of the files are being loaded. Additionally, you can see if anyone has any of the files checked out. Other possible operations are described below.

**Figure: Agile Load Preview**




**Note** Even if files are checked out by someone else, they will be loaded using “Get” (e.g. load without checkout), so that you always have full access to the files necessary to bring up the model in CAD.

**Figure: Load Dialog Toolbars**

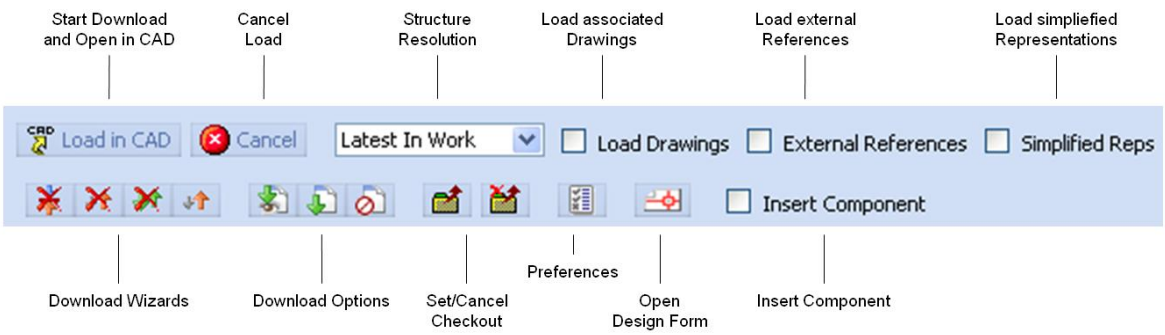
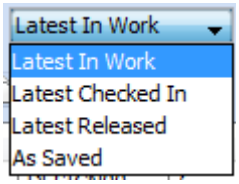



**Table: Toolbar Options**

| | |
|---|---|
| **Load in CAD Button** | Start the download of files and open the files in CAD depending on the selected download options. |
| **Cancel** | Cancel Load Preview and unlock CAD. |
| **Structure Resolution** | This selector defines the structure resolution used for retrieving components of Assemblies in PLM. The structure resolution will rerun if this option is changed. The default can be defined in the preferences *Structure Resolution on Load*.<br>  <br>**Latest In Work** – Use the latest version of a child design object.<br>**Latest Checked In** – Use the latest checked in version of a child design object.<br>**Latest Released** – Use the latest version of a child design, which is attached to a released Part.<br>**As Saved** – Use the child version from when the assembly was saved in PLM. |
| **Load Drawings** | If checked, the associated drawings are added to the list of files to load, by looking up the "where used" of the Designs. The structure resolution will rerun if this option is changed. |
| **External references** | If checked, the relationships table is scanned for external references, and if any are found they are added to the list of files to load. The structure resolution will rerun if this option is changed. |
| **Simplified Reps** | If checked, simplified representations are loaded to reduce CAD memory consumption. This feature is only supported with Pro/ENGINEER. |

| | |
|---|---|
| **Download Wizards** | These buttons provide a smart selection logic depending on the file status.<br>  Discard all local changes, where a newer version is available in PLM. The newer version is downloaded from PLM.<br>  Discards all local changes and reloads the version from PLM.<br>  Reload all local files from PLM, even if they are up to date.<br>  Keep all local changes, no local changes are discarded. |
| **Download Options** | Sets the desired load option for selected components. Determines how the file will be loaded from PLM to local disk. These options are automatically set by the system but they can be overridden by the user.<br>  Download the file to disk and display file in a CAD window<br>  Download the file to disk<br>  Do not download file to disk |
| **Set Checkout** | Set checkout reservation for the selected components. |
| **Cancel Checkout** | Cancel checkout reservation for the selected components. |
| **Preferences Button** | The preferences form pops out of the left sidebar. |
| **Open Design Form** | Opens the Design object form in Agile Web Client. |
| **Insert Component** | If an Assembly is active in CAD, the element selected for load is inserted into the Assembly structure in CAD, rather than opened in a separate window. The capabilities and positioning dialogs depend on the CAD system. |
| **Add all Drawings** | Checks all components within the Load dialog, and adds any related drawings to the Load dialog. |

**Figure: Load Dialog Object Table**

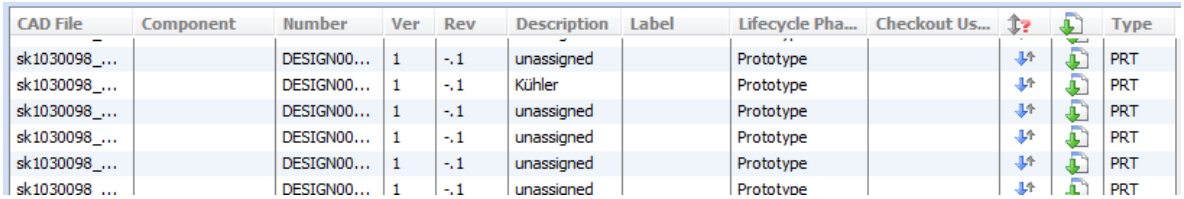
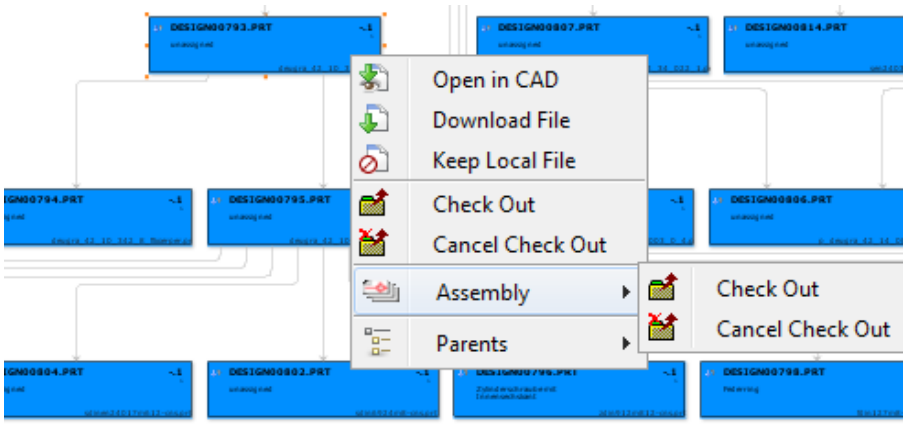



**Table: Fields in the Load Dialog**

| | |
|---|---|
| **CAD Object** | The CAD file name. |
| **Component** | Component Type of the Design object in Agile. |
| **Number** | Number of the Design object in Agile. |
| **Version** | Version of the Design object. This number starts at 1 and increments by 1 for each check in. |
| **Rev** | The Revision of the Design object, which includes a major and minor component. Each version has a unique revision, which is controlled by the part revision logic. |
| **Description** | Description of the Document object in Agile. |
| **Label** | Label for the specific version of the Design object. |
| **Lifecycle** | The lifecycle phase associated with the current Design version. |
| **Checkout User** | Current checkout user, if any. |
| **[File Status]** | Gives the status of the particular file, both in PLM and on the local disk. |
|  | The local file is exactly the same as what is in PLM. |
|  | The file has been updated in PLM, and so is more recent than what is on local disk. |
|  | The file has been modified locally, and so is more recent than what is in PLM. |
|  | The file has been modified both on the local disk and in PLM. |
|  | The file status cannot be determined. |
| **[Download Option]** | Determines how the file will be loaded from PLM to local disk. |
|  | Load and display in a CAD window. |
|  | Load but don't display. |
|  | Do not load. |
| **CAD Type** | Shows the CAD file extension, which can be used for sorting. |

# Multi-Select and Context Menus

Since you will often have many items listed in the Load dialog, it is convenient to be able to set options for multiple items at a time. This is made possible by multi-select and context menus.




Since the structure information is known, the *Assembly* sub-menu provides recursive functionality to perform operations for all the children of the selected assembly.

To multi-select, simply click within any item in the window and either hold down on the left mouse button and drag the cursor, or use Shift-click or Control-click. Once you have selected the desired items, you can use the context menu (right mouse button) to execute any of the commands listed in the following table.

**Table: Load Dialog – Context Menu**

| Command | Description |
|---|---|
| **Open in CAD** | Download and open the selected file in CAD. |
| **Download File** | Sets selected files to be retrieved to local disk. |
| **Keep local File** | Sets selected files to NOT be retrieved to local disk. |
| **Check Out** | Sets checkout reservation for currently selected objects. |
| **Cancel Check Out** | Cancels checkout reservation for currently selected objects. |
| **Assembly submenu** | Performs contained actions recursively for the selected object and all children. |

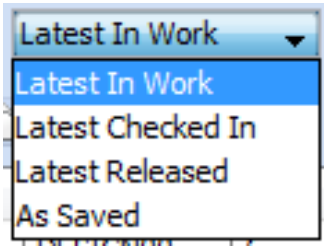
| | |
|---|---|
| **Parents submenu** | Performs contained actions recursively for the selected object and all its parents. |

# Structure Resolution Options

The EC Web Connector provides full capability for loading any desired revision of the CAD files. This is done through a combination of two controls within the EC Web Connector UI:

**Structure Resolution Select Box:** The Structure Resolution select box has four choices, *Latest In Work, Latest Checked In*, *Latest Published* and *As Saved*.

**Figure: Structure Resolution Select Box**




The logic of these options is:

**Latest In Work** – Starting with the selected object, each Design version in the structure being checked out by you will be chosen to load. If there is no checked out version for a given Design, or if it is checked out by someone else the latest checked in version is chosen instead. This is the most up-to-date view you can get of the CAD design. It loads the latest version of each sub-assembly and component that can be extracted from the tree.

**Latest Checked In** – Starting with the selected object, each checked-in Design version in the structure is chosen to load. This is a good option when loading a CAD design from PLM in order to make a change, while ensuring that the latest version of all sub-assemblies and components are being loaded. Note that these versions may be more up-to-date than the versions with which the top assembly was actually saved.

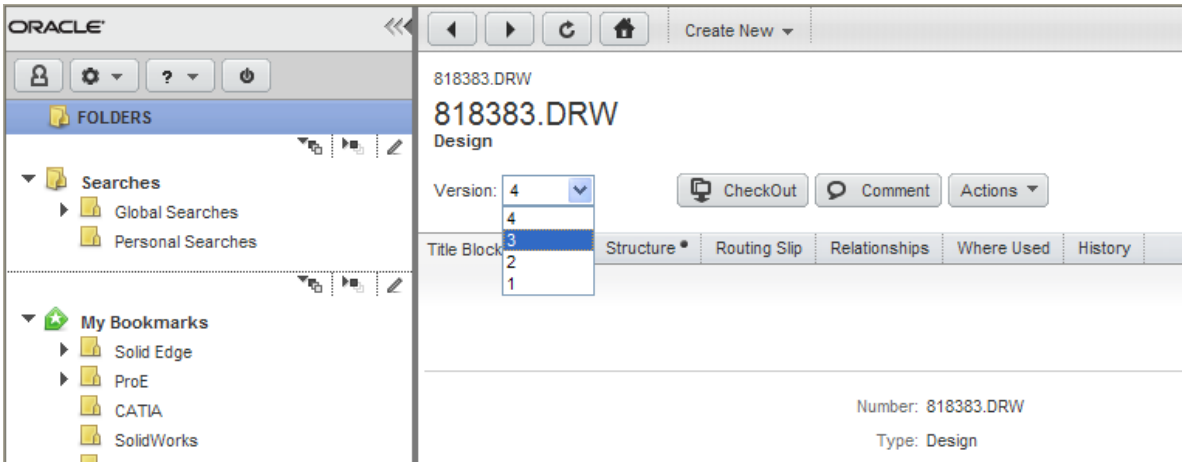
**Latest Released** – Starting with the selected object, each Design version in the structure being attached to a released part is chosen to load. This is a good option when loading a CAD design out of PLM, in order to get a released assembly structure and ensuring that the latest released version of all sub-assemblies and components are loaded as well. Note that these versions may be more up-to-date than the versions with which the top assembly was actually saved.

**As Saved –** Starting with the selected object, each Design in the structure will be loaded just the way it was saved into Agile. This is known as "Fixed" structure resolution, and that is what is shown in the Design Structure tab. This is useful for loading any version, current or past,

just the way it was saved into PLM.

Note that with any of these options, if you want to load a past revision, you must execute the load directly from the Design object web form, not from the search results list. That is necessary because you need to be able to select the version of the top Design, as shown below.

**Figure: Version Selector – For use with "As Saved"**




Regardless of which load option you use, the Load dialog will always display the revision that is being loaded for each item. If this does not appear to be correct, it is always possible to cancel the process from the Load Dialog. Nothing will be loaded in this case.

# Creating New Objects

## Introduction

The Create Object frame is a dialog frame that enables the user to create new objects simultaneously in CAD and Agile9. The object creation is based on template files stored in PLM, the user can choose the template being used. The dialog can be opened with the *New* command in the ribbon or menu bar of the user's CAD. Depending on the CAD this command may appear in a submenu called *Agile*.

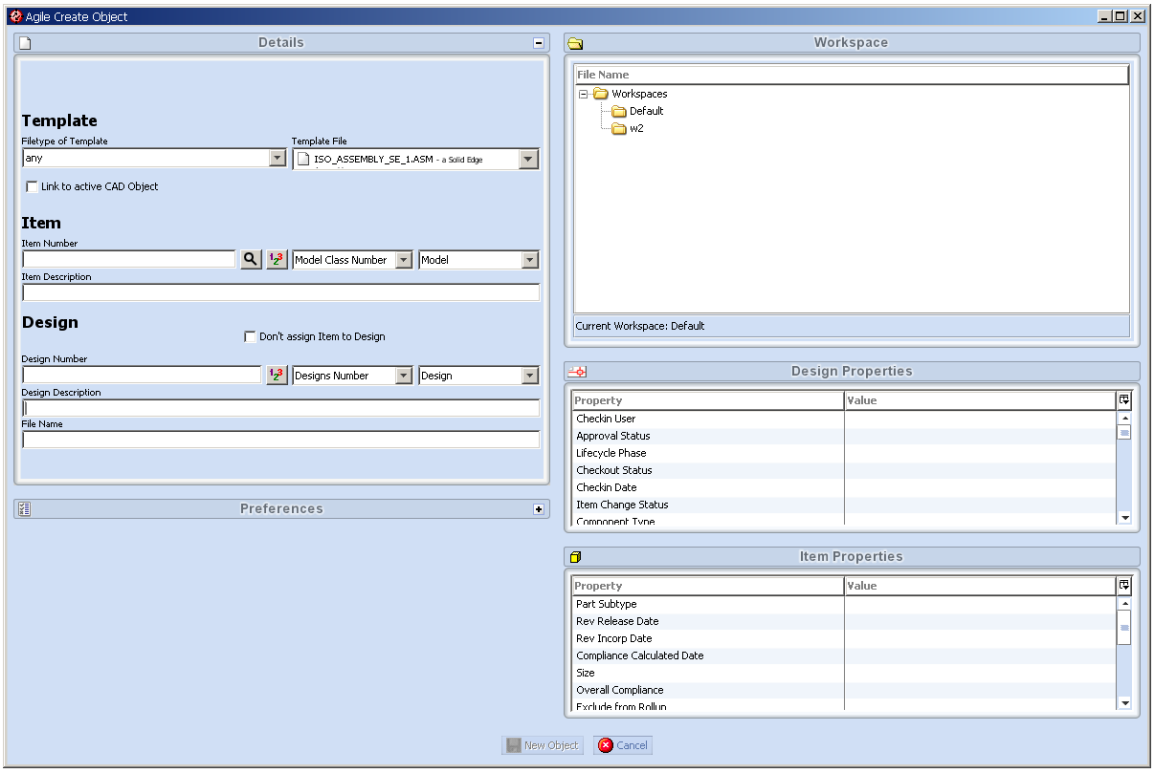
The basic workflow for creating new objects includes the following steps:

- Selection of a template using the Template File combo box
- Assignment of a design subclass to the object using the Design Subclass combo box (optional, the class having been pre-selected on template selection could be used as well)
- Assignment of a design number using the Design Autonumbers combo box and the Design Autonumbers button or manual input
- Assignment of an item (optional)
- If every necessary parameter is correctly entered the New Object button becomes clickable and can be used to finish object creation. That means, the selected template file will be copied to the previously selected workspace directory on the user's machine and opened in the user's CAD. A corresponding Design object in PLM will be created simultaneously.

# Using the Create Object Dialog

The Create Object frame is subdivided into five sections. The first and most important among them is the *Details pane* on the left hand side. This component contains the basic elements for object creation.

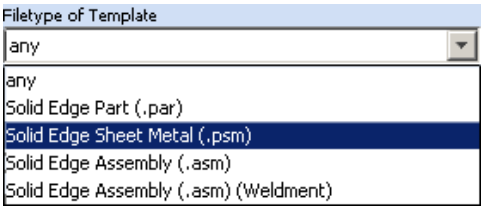
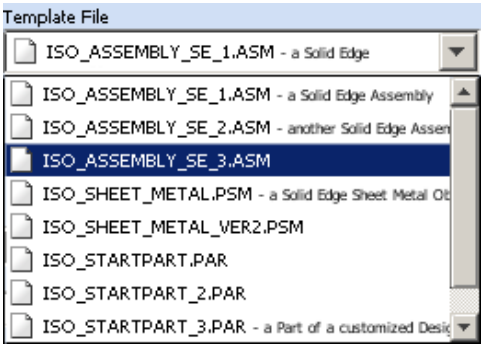
**Figure: Create Object frame, used to create new objects**




Other components in the Create Object frame are the *Preferences Pane* (left hand side, collapsed in the figure*)*, the *Workspace Pane* and the *Design* and *Item Properties* tables on the right hand side. These four components behave similar to the components used in the sidebar of the Save and Load Dialog, however, there are minor differences. For a detailed documentation of these components, please consult the appropriate sections for the Save and Load Dialog in this document. The following paragraphs explain the Details pane and the differences between the components used in the Create Object frame in comparison to the same components used in the Load respectively Save Dialog.

## Details Pane

**Table: User Interface Elements of the Details pane**

| Element | Description |
|---|---|
| **Filetype of Template combo box**<br>  | Lets the user choose a file type for the templates displayed in the Template File combo box. The subtype is displayed in brackets behind its file type if templates with assigned subtypes are available for selection. If the *any* option is selected, all available templates will be displayed. If an entry without a subtype is selected, all templates of that file type are displayed (no matter if they have a subtype assigned or not), if an entry with both a file type and a subtype is selected, only the templates of that file type **and** subtype are displayed. |
| **Template File combo box**<br>  | Lets the user choose a template file from which a new CAD file can be created. The combo box displays the names of all suitable template files for the file type (or file type and subtype) selected in the Filetype of Template combo box and their description (if given in PLM).<br><br>**Note:** New objects can only be created if a valid file is selected in this combo box. |
| **Item Subclass combo box** | In this combo box the subclass of an item object in Agile9, which can be optionally created, can be chosen. This combo box is set inactive if no item should be created. |
| **Item Autonumber combo box** | This combo box contains all available autonumber generators for the item subclass currently selected in the Item Subclass combo box. |
| **Item Autonumber button**<br>  | If this button is clicked, the next available number from the selected autonumber generator will be inserted into the Item Number text field. |

| | |
|---|---|
| **Item Search button**  | This button executes a part search in PLM using the values entered in the Item Number text field and the Item Description text field. If there is one match, the found number is written into the Item Number text field. If more than one match is found, a dialog pops up and the user can select the desired part from the list of found parts. This function can be used to assign a design to an existing part. |
| **Item Number text field** | This text field is used to enter the item number of an item object in Agile9 that should be linked to the design object in case an item should be created at all. |
| **Item Description text field** | The description of the part item that will be assigned to the design. |
| **Design Subclass combo box** | A combo box that enables the user to choose a design subclass for the object being created. When a template is selected in the Template File combo box, the pre-assigned design subclass of that template is pre-selected in this combo box. However, the user can choose every other available subclass if he wishes. |
| **Item Description text field** | The description of the part item that will be assigned to the design. |
| **Design Autonumber combo box** | This combo box contains all available autonumber generators for the design subclass currently selected in the Design Subclass combo box. |
| **Design Autonumber button**  | If this button is clicked, the next available number from the selected autonumber generator will be inserted into the Item Number text field. |
| **Design Number text field** | This text field is used to enter the design number of the design object in Agile9 that is being created simultaneously with the CAD file.<br><br>**Note:** New objects can only be created if this text field is not empty. |
| **Design Description text field** | The value that will be used as the description assigned to the design object that is being created. |

| | |
|---|---|
| **File Name text field** | The CAD file name that is being saved into Agile. |
| **New Object button** | Creates a new object from the template being selected in the Template File combo box in CAD and creates a structure for that object in Agile9.<br><br>**Note:** This button cannot be activated as long as no template is selected and no design number is assigned due to the matter of fact, that these are necessary parameters. |
| **Cancel button** | Closes the Create Object frame without changing anything. |

## Preferences Pane (initially collapsed)

The Preferences Pane behaves much like the Preferences Pane in Save or Load Dialogs. The only difference is that changes in the Class Default Settings are automatically applied to the corresponding elements in the Details Pane, too. In detail that concerns the Design Subclass combo box, the Item Subclass combo box and the Autonumbers combo boxes belonging to these. As well as the Details Pane the Preferences Pane can be collapsed to save screen space if needed. For further information on the Preferences Pane please refer to the paragraph *Preferences Dialog* in this document.

## Workspace Pane

The Workspace Pane is used to select the workspace folder, there are no differences in behavior compared to Save and Load Dialogs. For further information on how to work with workspaces please consult the chapter *Workspace Management* in this documentation.

## Design and Item Properties Tables

These two properties tables are used to display design and item properties of the object that should be created. The "Value" column is usually editable and a few entries of the Design Properties may be preassigned depending on the selected template in the Template File combo box. A property will be preassigned if a corresponding entry in the Agile form of the corresponding template object exists. Whenever a new template is selected in the combo box the predefined values of that template will be loaded into the table. Above that there are no differences to the properties tables used in the sidebar of the Save and Load Dialogs.

# Workspace Management

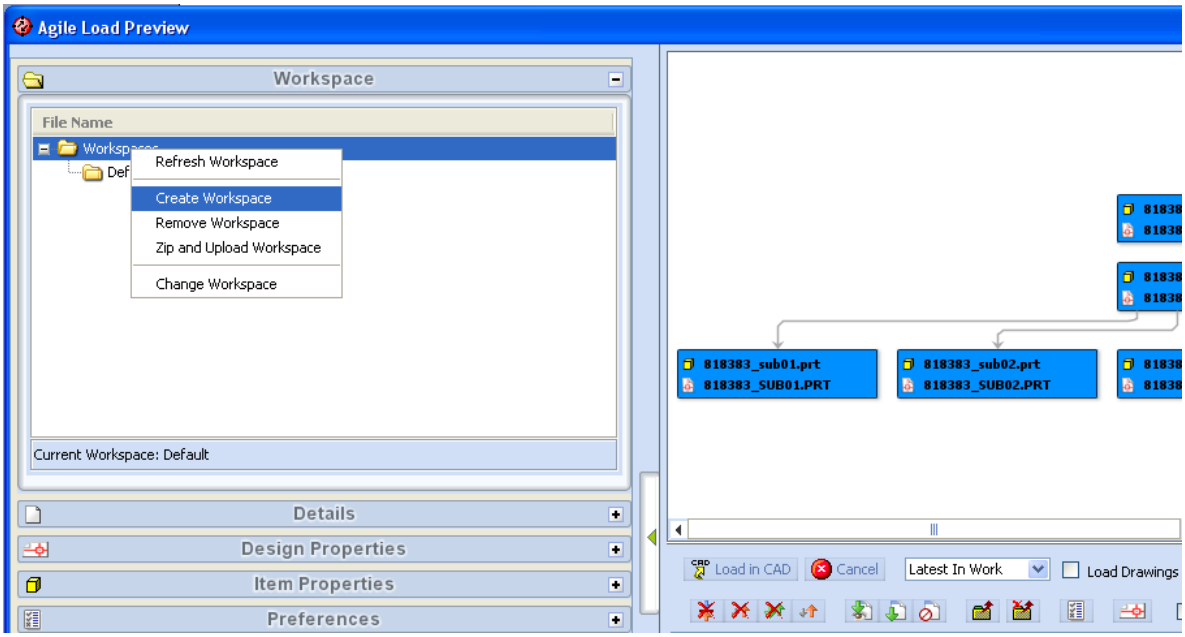
## Introduction

In order to control updates to CAD files and the Design structures in Agile, the CAD Connectors provide the ability to manage the change process using the inherent capability in Agile. There are primary components to this capability:

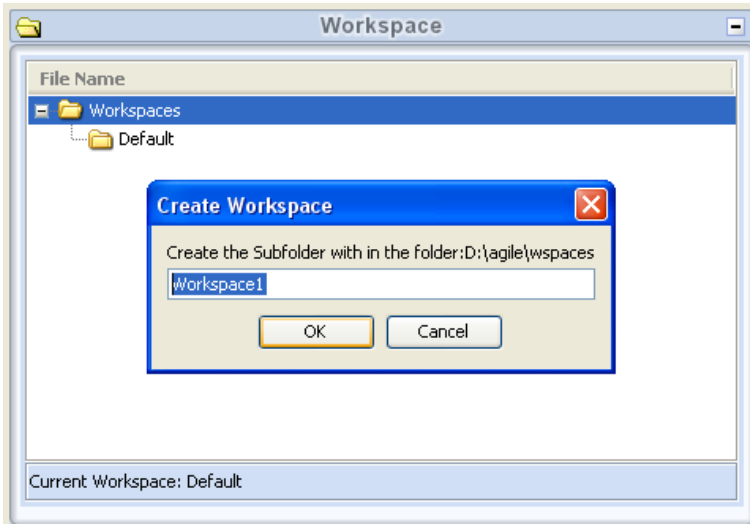
- Controlling the ability to update the files in Agile, using privileges, checkout reservation, and versioning.
- Controlling the change process workflow, using revisions and ECOs.

# Set Workspace During Load, Save or Object Creation

Expanding the left Sidebar in the Load or Save Preview you can select the Workspace area by clicking the plus sign beside the Workspace label. In the Create New frame the Workspace pane can be accessed directly. The Workspace pane enables you to create, set or delete Workspaces, which correspond to folders on your local disk.




You can create new workspaces using the context menu *Create Workspace* in the workspace tree structured view:

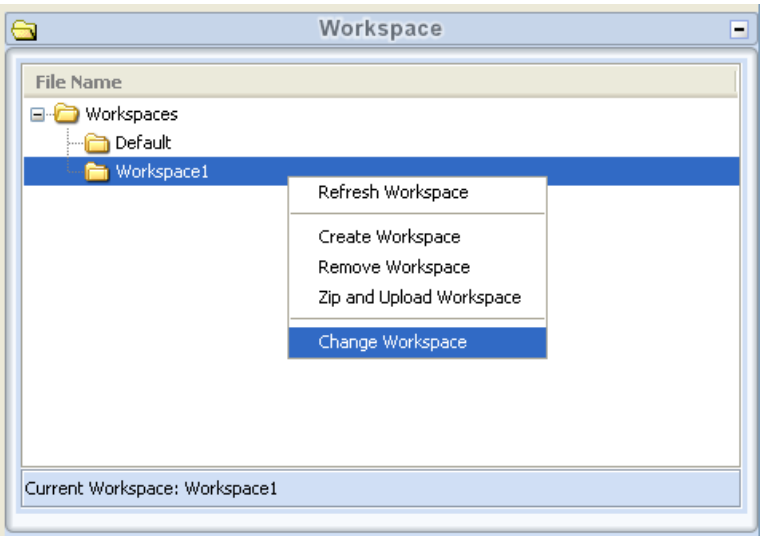


To set the Workspace to be used for load, save or object creation you select the *Change Workspace* context menu pick. The current workspace is shown at the bottom of the Workspace pane.

During load all downloaded files will be copied to the current selected workspace and loaded from there into CAD. Make sure your CAD environment is setup properly to pull the component files out of the same directory as the assembly.

During save, all files selected for save will be copied to the current workspace. After changing the workspace the save of complete trees to new workspaces is recommended.

During object creation the newly created CAD file is saved to the current workspace as well.




## Archive and Zip a Complete Workspace

The *Zip and Upload* command can be used to compress a complete workspace, including all files and subdirectories. The zip file will be uploaded to PLM in a Design object named `Workspace_{username}_{timestamp}`. The created folder will be linked to the bookmarks of the current login user.

The zipped file also contains the PLM cache information and can be used to save or transfer complete workspaces, including all CAD files and PLM related information. When extracted into a workspace on a different machine, the Workspace Manager imports the **cache.xml** file and can detect the correct PLM objects and versions corresponding to the local CAD files. **It is recommended that you extract complete workspaces into a new empty target workspace.**

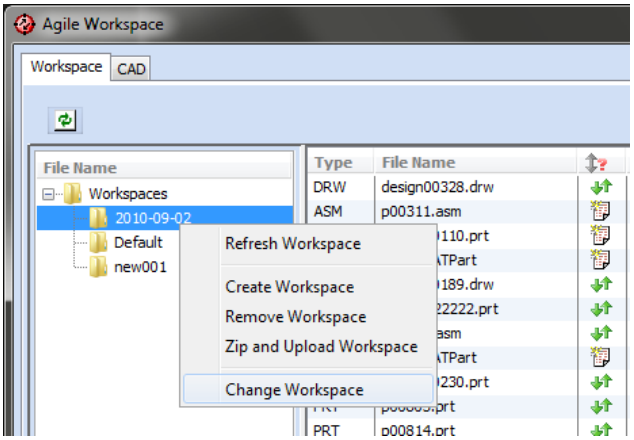
# CAD working directory vs. EC Workspaces

Be aware that changing the working directory of the CAD system like Pro/ENGINEER and changing the EC Workspace are two different things and lead to different behaviour of the integration.

The working directory is the active directory in the CAD system, where the engineer is working. This directory is not necessarily the same directory where the CAD files loaded in session are located, because after load the working directory may be changed or some components are loaded from external library paths.

The EC workspace is a managed directory of the MCAD integration that can be displayed in the Workspace Manager. EC workspaces are located in a dedicated local area on disk defined by the environment setting CAX_WORKSPACE_ROOT. All directories below this root can be displayed in the EC Workspace Manager. External directories outside this root are not displayed, but the connector tracks the file status of any file independent from the path where it resides.

The best practice is to have the CAD working directory and EC Workspace set equal. This can be done using the *Change Workspace* command in the Workspace Manager. This also sets the current working directory in Pro/ENGINEER to the same value.



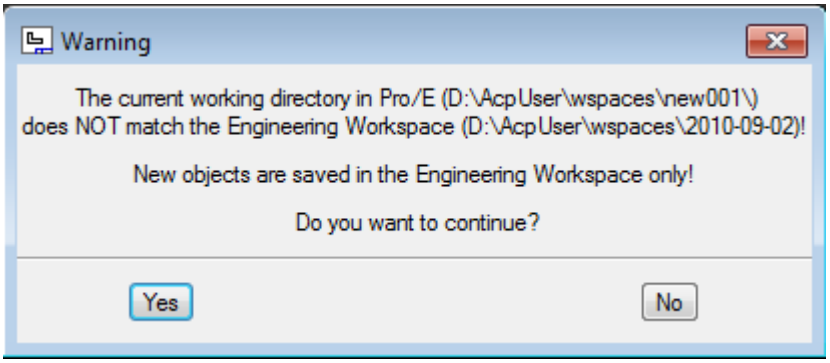
**Attention: If the current working directory is changed in CAD manually, the EC Workspace is NOT changed!**

# Workspaces on Save

In order to keep assembly integrity the location of files on disk are not changed during save. Only completely new files not stored on disk are saved into the active EC workspace.

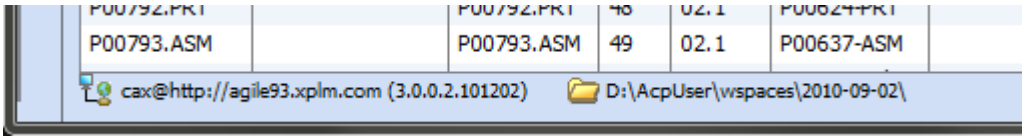
**Attention: Setting the current working directory in CAD has NO effect on where the new files go. Set the EC workspace to control where new files go during save to Agile.**

If the current working directory in Pro/Engineer doesn't match the active EC workspace a warning message is displayed to the user to prevent the user saving new files to a different location than expected.




The user can now decide to continue and create new objects into the Engineering Workspace folder or to abort saving and reset the engineering workspace and the CAD working directory.

The current selected workspace can be seen in the bottom of the Save Preview Window.

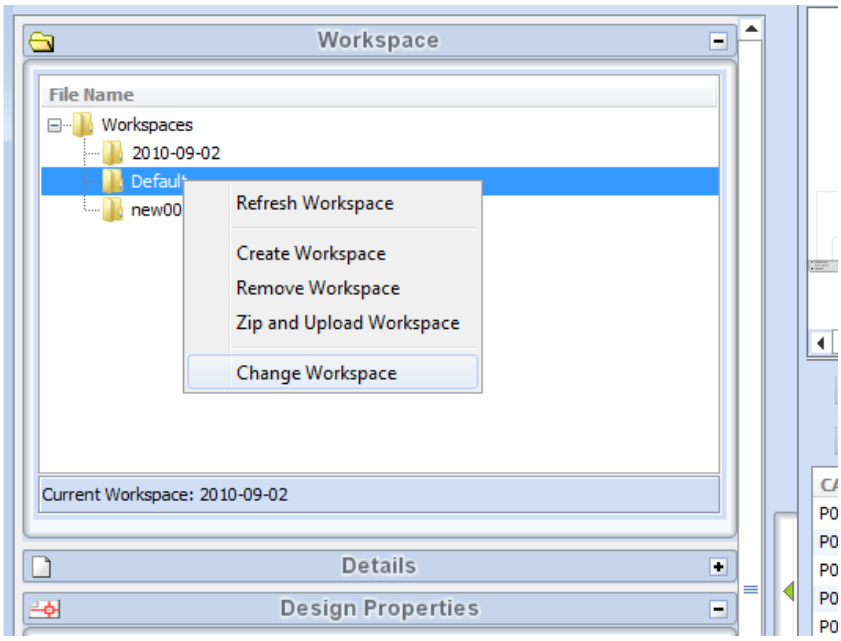



Even in the Save Preview the engineering workspace can be changed in the sidebar. **Changing it in the Save Preview will NOT reset the CAD working directory.** The current working directory in CAD is retained during save. It is reset to the same directory after saving as it was before saving. **If you intend to change both at once, use the Agile Workspace Manager before saving.**

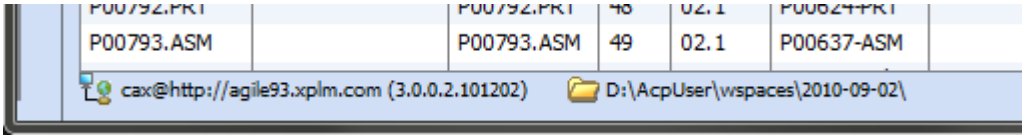
If the CAD files are renamed during initial save and the file location is changed, the original file is retained. Viewables are generated in the active workspace of the CAD file.

## Workspaces on Load

On load from Agile the target workspace can be selected using the *Change Workspace* command in the Workspace Manager or in the sidebar of the Load preview as shown below. On change of the current workspace, the file status checks are executed and the display is refreshed. Both commands also set the current working directory in CAD.




**ATTENTION: Before changing the workspace, it is recommended that you clean the CAD session.** The current selected workspace can be seen in the bottom of the Load Preview window.




# Workspaces on CAD Start

The last workspace is stored in the **connect.properties** file in users AgileCache directory. During startup Pro/ENGINEER scans this file for the workspace and sets the CAD working directory to the last workspace.

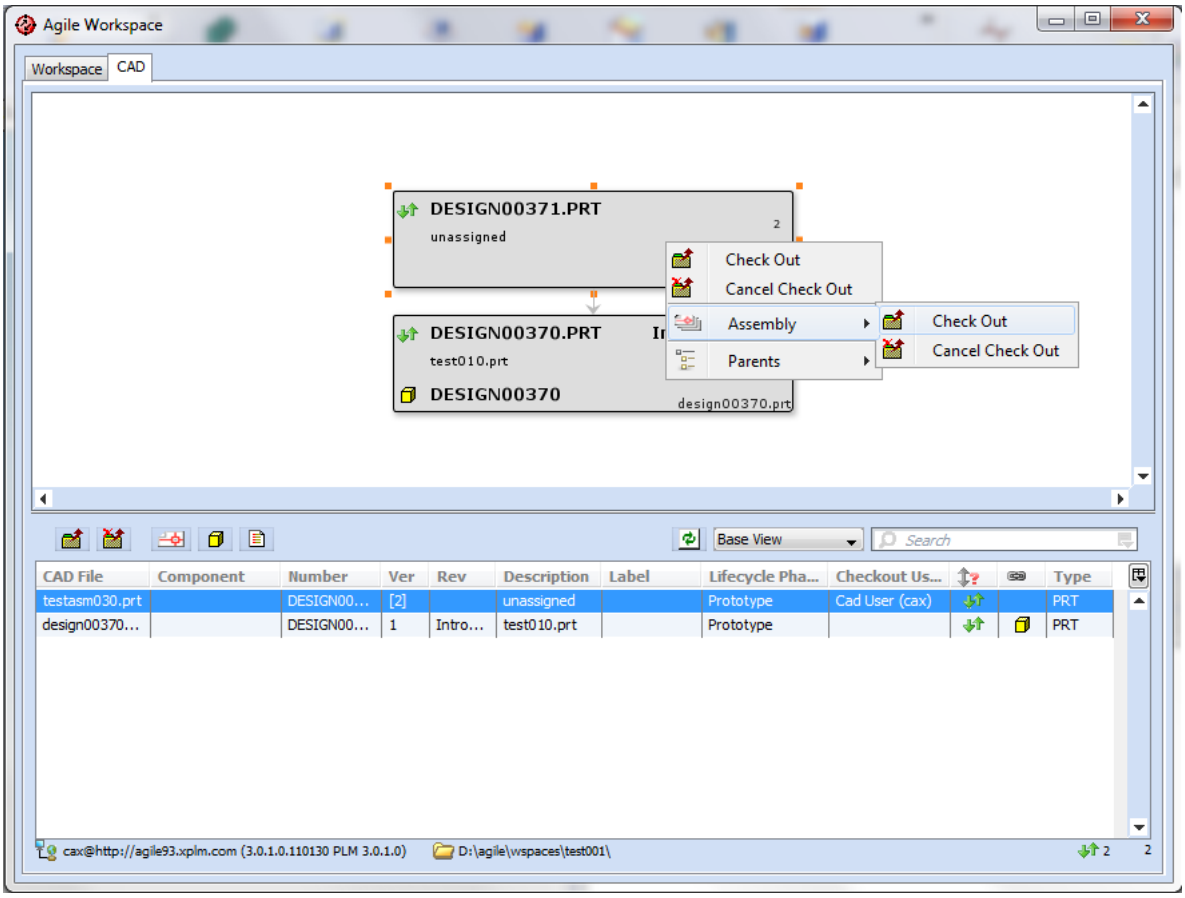
# Using the Workspace Manager

The *Agile →Workspace Manager* command is used to view and change the checkout status of the CAD files controlled by Agile and for the managed workspaces. The following figures and tables explain the commands available; how you use them to effectively manage the change process for the CAD designs is described following that.

# Using the Workspace Manager CAD Session Tab

After launching the Workspace Manager from CAD, the content of the current CAD session is shown in the CAD tab. Here you can perform PLM actions on the designs that are active in your CAD session.

**Figure: Workspace Manager - CAD Session Tab**



Details of the CAD Session tab dialog are described in the tables below.


**Table: Options of the CAD Session Toolbar**

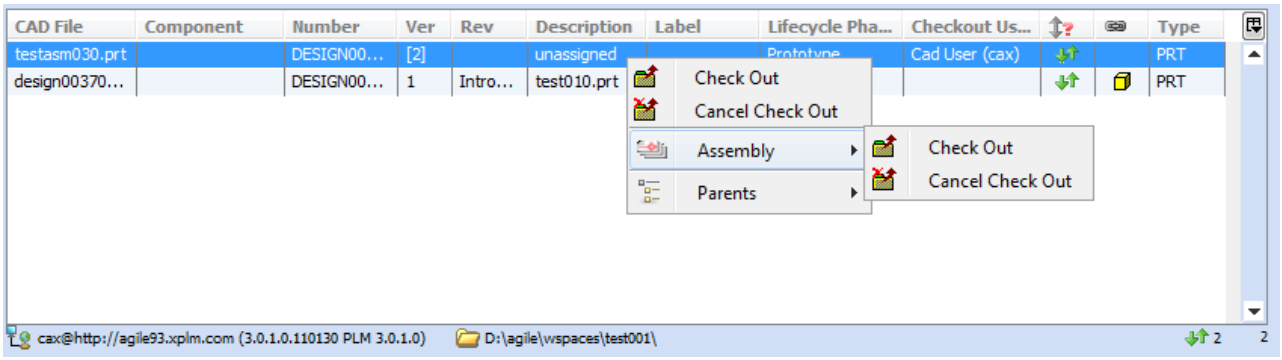
| | |
|---|---|
| Check Out | Sets checkout reservation for currently selected objects. |
| Cancel Check Out | Cancels checkout reservation for currently selected objects. |
| Open Form for Design, Part or Change | Open the desired form in Web Client. |
| Refresh | Updates the display from Agile. |

**Table: List Fields and Controls of the CAD Session Tab**

| | |
|---|---|
| **CAD Object** | The CAD filename. |
| **Component** | Component Type of the Design object in Agile. |
| **Number** | Number of the Design object in Agile. |
| **Ver** | Version of the Design object. |
| **Rev** | The Revision of the Design object, which includes a major and minor component. Each version has a unique Revision, which is controlled by the part revision logic. |
| **Description** | Description of the Design object in Agile. |
| **Label** | Label for the specific version of the Design object. |
| **Lifecycle Phase** | Lifecycle phase of the specific version. |
| **Checkout User** | The name of the current checkout user, if any. |
| **[File Status]** | Gives the status of the particular file, both in PLM and on the local disk. |
|  | The local file is exactly the same as what is in PLM. |
|  | The file has been updated in PLM, and so is more recent than what is on local disk. |
|  | The file has been modified locally, and so is more recent than what is in PLM. |
|  | The file has been modified both on the local disk and in PLM. |
| **[Part Assignment]**  | Indicates whether the file has a part assigned in Agile. If so, the part icon will be displayed in the column. |

| | |
|---|---|
| **CAD Type** | Shows the CAD file extension, which can be used for sorting. |

## Multi-Select and Context Menus




You can multi-select within the dialog to operate on more than one item at a time. Once you have selected the desired items, you can use the context menu (right mouse button) to execute any of the commands listed in the table below.

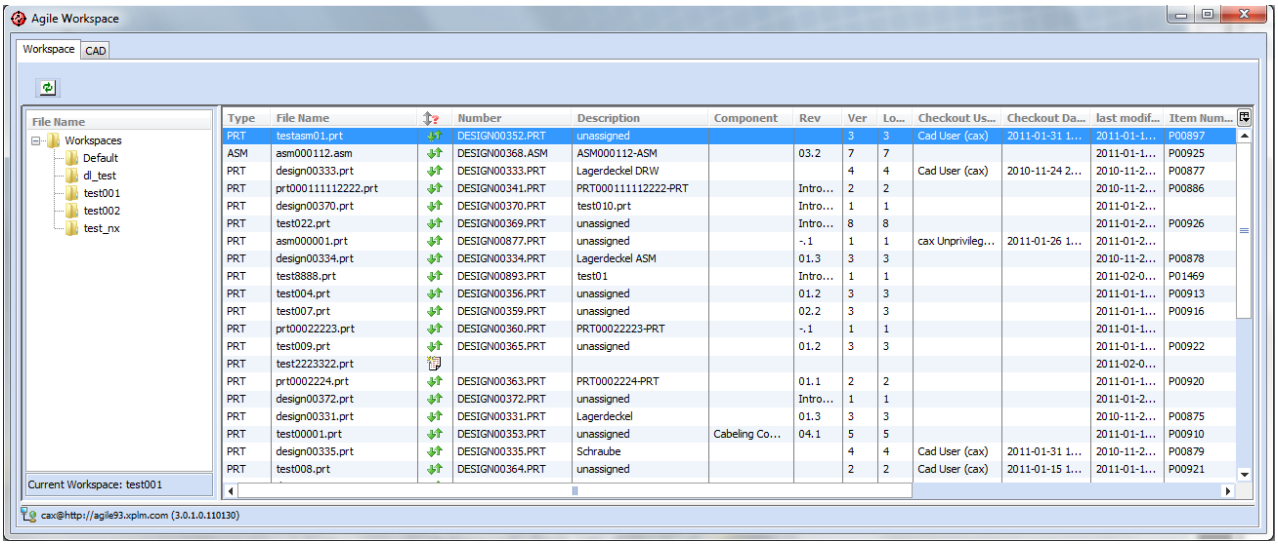
**Table: Context Menus of CAD Session Tab**

| Command | Description |
|---|---|
| **Check Out** | Sets checkout reservation for currently selected objects. |
| **Cancel Check Out** | Cancels checkout reservation for currently selected objects. |
| **Assembly submenu** | Performs contained actions recursively for the selected objects and all their children. |
| **Parents submenu** | Performs contained actions recursively for the selected objects and all their parents. |

# Using the Workspace Manager Workspaces Tab

The Workspaces tab is a CAD filtered view on your local disk and workspaces structure. For the CAD files, it displays the PLM information for each single file. The file status column displays the status of each local file, even if it is not loaded in CAD.

**Figure: Workspace Manager – Workspaces Tab**



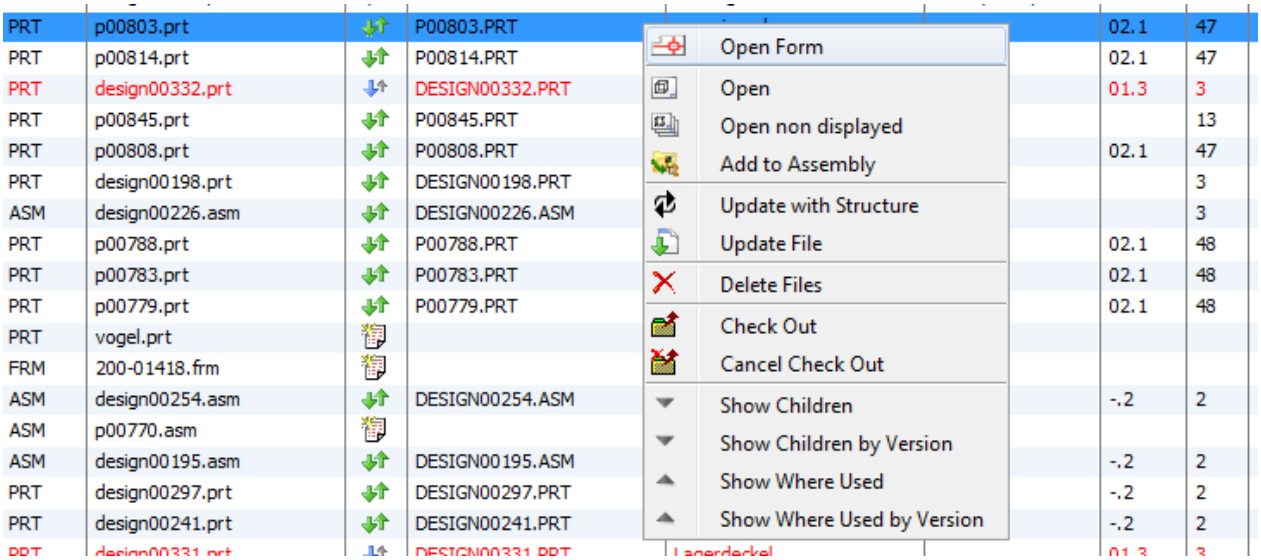

Details of the Workspaces tab dialog are described in the tables below.

**Table: List Fields and Controls of the Workspaces Tab**

| | |
|---|---|
| **CAD Type** | Shows the CAD file extension, which can be used for sorting. |
| **File Name** | Local CAD filename. |
| **[File Status]** | Gives the status of the particular file, both in PLM and on the local disk. |
|  | The local file is exactly the same as what is in PLM. |
|  | The file has been updated in PLM, and so is more recent than what is on local disk. |
|  | The file has been modified locally, and so is more recent than what is in PLM. |
|  | The file has been modified both on the local disk and in PLM. |
| **Number** | Number of the Design object in Agile. |
| **Description** | Description of the Design object in Agile. |

| | |
|---|---|
| **Component** | Component Type of the Design object in Agile. |
| **Revision** | The Revision of the Design object, which includes a major and minor component. Each version has a unique revision, which is controlled by the type of Label assigned. |
| **PLM Version** | Latest version of the design in PLM. |
| **Local Version** | Corresponding PLM Version of the local file. |
| **Date** | An optional aate field, which can be associated to the version of the Design object. |
| **Checkout User** | The name of the current checkout user, if any. |
| **Checkout Date** | The date of the current checkout, if any. |
| **Modify Date** | Local modification date. |
| **Item Number** | The number of a related Part object in PLM. |
| **Rev** | The revision of the Part object. |
| **Path** | Full path information of the file. |
| **Size** | Local file size. |

## Multi-Select and Context Menus




You can multi-select within the dialog (CTRL+left mouse), to operate on more than one item at a time. Once you have selected the desired items, you can use the context menu (right mouse

button) to execute any of the commands listed in the table below.

**Table: Context Menus of Workspace Manager**

| Command | Description |
|---|---|
| **Open Form** | Open the Design form in Web Client. |
| **Open** | Open the selected file from disk in CAD and display it in a CAD window. |
| **Open non displayed** | Open the selected file from disk in CAD Session, but do not display a CAD window. **This is available for Pro/E and NX only.** |
| **Add to Assembly** | Open the selected file from disk in CAD and launch the add component dialog in CAD. **This is supported for Pro/E, Solid Edge and CATIA.** |
| **Update with Structure** | Performs a structure resolution in PLM and displays a load preview. |
| **Update File** | Updates the local file with the latest version from PLM. No structure resolution is executed to retrieve components. |
| **Copy Files** | Copies all selected files to a virtual clipboard, note that this is not the system clipboard. |
| **Cut Files** | Copies all selected files to a virtual clipboard. The selected files can then be removed from their original directory and moved to another directory using the *Paste Files* function. As long as *Paste Files* is not executed, the selected files remain visible. |
| **Paste Files** | Inserts the files from the virtual clipboard to the workspace directory on which the *Paste Files* function is executed. If the files were sent to the clipboard with the *Cut Files* function, they will be removed from their original directory and the clipboard. However, they remain in place and on the clipboard if the *Copy Files* function was used.<br>If there are files with similar file names in the destination directory the user is promted for overwriting them. If the user denies, then the previously selected files will not be removed even if the *Cut Files* function was used on them. Pasting files will not work if the files to be processed are removed from the source directory before the *Paste Files* function is used. |
| **Delete Files** | Delete selected files and all local version copies from the current workspace directory. |
| **Check Out** | Check out the selected Design in PLM. |
| **Cancel Check Out** | Remove the check out of the selected Design in PLM. |
| **Show Children** | Highlights the children of an Assembly in the workspace based on the current structure in PLM. |
| **Show Where Used** | Highlights the parents of a file in the workspace based on the where used structure in PLM. |

The *Open non displayed* command can be used for initial data load or to save multiple top level objects, like drawings, to PLM. The user can bring multiple objects into the CAD session and can then use the *Save Session* command in Pro/E to save everything in one save process.

**ATTENTION: In order to create PDF for drawings in Pro/E the drawing must be displayed and regenerated in session. Otherwise no PDF is created.**

# Understanding the Change Process

The Design object in Agile PLM was designed to have a simpler change process than for Items (Documents and Parts). The change process is completely self-contained within the Design object itself; there is no separate Change object (e.g. ECO) required. The elements of the change process within the Design object are as follows:

**Version** – The Version field is the basic tracking mechanism for changes to the Design. Each version of a Design object can have a unique set of files and a unique structure. The version is a sequential number.

**Revision** – The Revision field is used to track a revision code. The revision of assigned Parts is initially written to the Design object during the BOM publishing process. To fully support this process, it is necessary to download and install the *DesignRevision Groovy script* sample from the Oracle Samples website http://www.oracle.com/technology/sample_code/products/agile/9.3/index.html. This script is used to set key attributes such as the Design Revision in a release workflow, and can be easily modified to fit your company processes. The revision code contains a major revision and minor revision component (e.g. A1). Each version of a Design will have its own Revision code.

**Revision Date** – The Revision Date is an optional field to track a date associated with the revision. Each version of a Design can have its own revision date code.

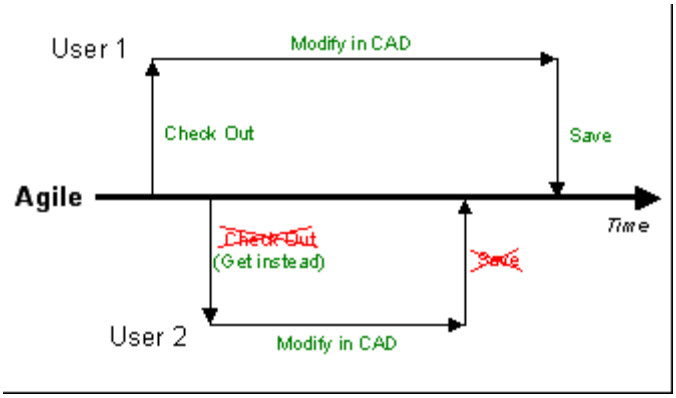
**Routing Slip** – The Routing Slip is a tab that is used to assign approvers and observers for any given version of a Design. Once assigned to a Routing Slip, the approvers and observers will be notified through the normal Agile notification methods, and then can approve or reject the specific version of the Design.

# Concurrent Engineering

The EC Web Connector is specifically designed to support concurrent engineering, the ability for multiple designers to work on different portions of the same overall CAD Assembly at the same time. The most important consideration for concurrent engineering is that the ongoing changes by the designers be managed such that the files in the central repository (Agile) remain valid and up-to-date.

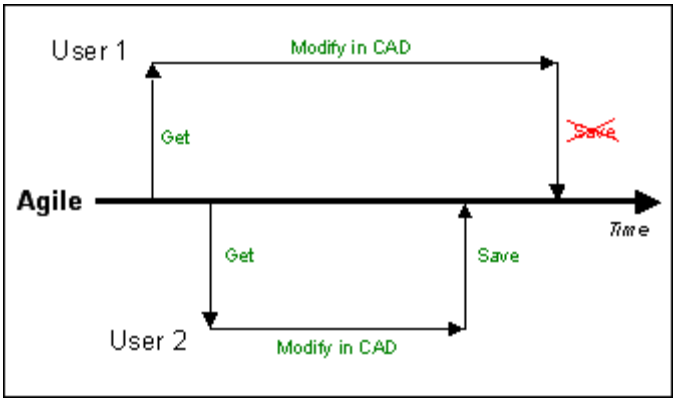
The EC Web Connector makes use of two basic control mechanisms to manage concurrent engineering: *Check out* and *timestamp*. Check out is a reservation mechanism inside Agile that is used to prevent other users from saving changes to something you are changing. Timestamp is a mechanism that relies on a timestamp value stored on each Design object in Agile each time you save. If your timestamp is up-to-date, meaning that no one has made a change more recently than when you loaded the file, then you can save into Agile. The following diagrams illustrate the fundamental scenarios involving Check Out and Versioning.

**Figure: Scenario 1 – Check Out Reservation**




- User1 loads file from Agile using check out, and proceeds to modify in CAD.
- User2 attempts to load file from Agile using check out, but cannot since User1 already has it reserved. The file is loaded using *Get* instead.
- User2 modifies in CAD, then tries to save into Agile. Save is denied since User1 has checkout reservation
- User1 then tries to save, which is successful.

**Figure: Versioning**




- User1 loads file from Agile using *Get*, and proceeds to modify in CAD.
- User2 also loads file from Agile using *Get*, and proceeds to modify in CAD.
- User2 tries to save into Agile, which is successful (because there is no check out reservation)
- User1 then tries to save into Agile. Save is denied due to out-of-date version (file in Agile has been updated since User1 loaded it).

| | |
|---|---|
| **Note** | Both Check Out and Version status can be checked from the **Workspace Manager** dialog. If a name is listed in the **Checkout User** field, then the item is checked out. If the **Changed in Agile** column is flagged, the item is out-of-date. |
| **Note** | All CAD files that are loaded from Agile into CAD are modifiable in the CAD system. That is, they are not loaded "Read-only". The ability to modify the files in CAD, does not necessarily mean that you will have the privileges to save into Agile. |

A preferences setting called **Check Out During Save** determines how the check out and version mechanisms are used to control concurrent engineering for your site. Mainly it affects the conditions under which saveing is allowed. If this option is set to *Force User Check Out*, then it is required to have check out reservation set for any Design that is to be saved. If it is set to *Automated Check Out*, then Designs can be saved and if not previously checked out, they will be automatically checked out prior to saving.

For most companies the use of *Force User Check Out* is recommended, in order to insure that users are aware when other users are working on models. In addition, it is recommended to set the check out on the Design as early as possible, either when using the Load command or by using the Workspace Manager during your CAD design modification.

**Table: Agile Save Option**

| | Value of Save Option | |
|---|---|---|
| Checkout Status | Automated Checkout | Force User Checkout |
| Checked out by someone else | Do not allow Save | Do not allow Save |
| Not checked out, and out-of-date | Allow Save | Do not allow Save |
| Not checked out, but up-to-date | Allow Save | Do not allow Save |
| Checked out by current user | Allow Save | Allow Save |

# BOM Publishing

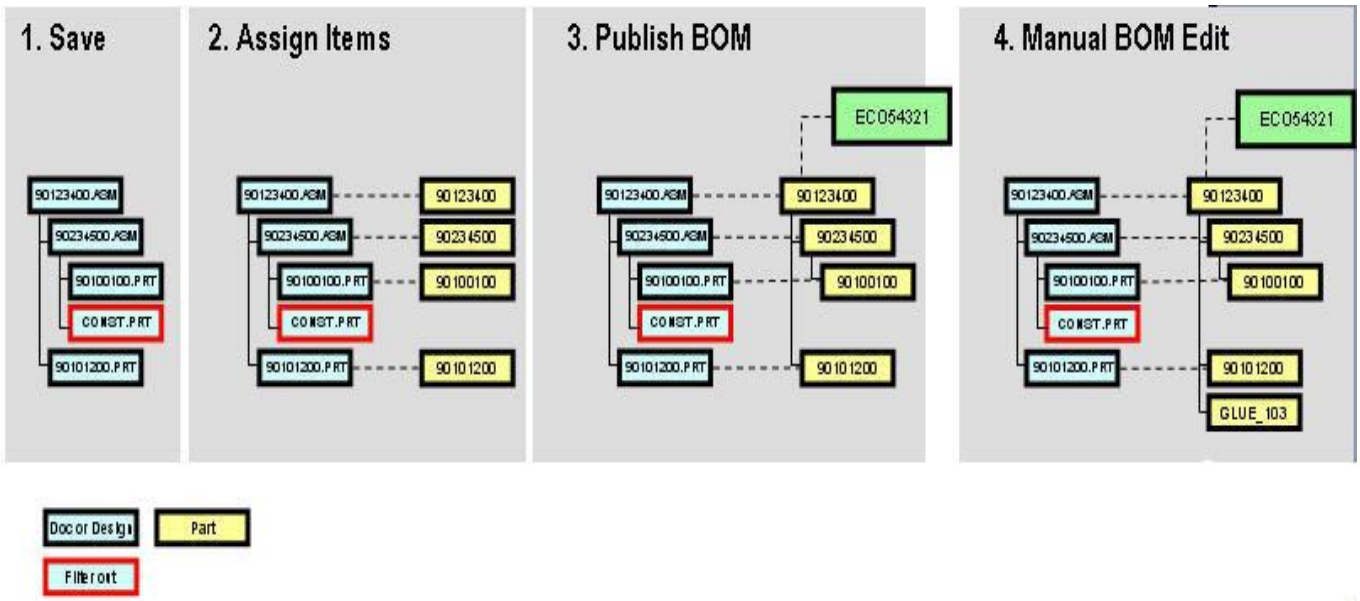
## Introduction

BOM Publishing, using the *Agile → Save* command, is used to create or update Agile Product Structures based on CAD Design Structures. The Product Structure, or "Part BOM", is the definition of your product that is passed to manufacturing. Since in many cases this structure closely resembles the structure of your CAD design, the BOM Publishing step can leverage this to decrease effort and increase accuracy.

## Overview of the BOM Publishing Process

The overall BOM Publishing process has four main steps, which are illustrated below. The first three steps can all be managed within only one save command depending on the preferences settings for the publish and part creation behavior. (See the *Item and Publish Preferences section* for details)

**Figure: The BOM Publishing Process**




**Save** – This step saves the Design structure into Agile PLM, using the *Agile → Save* command.

**Assign Items** – This step links each Design object to a corresponding item (typically a Part class object). This linking is done based on the type of mapping defined in your EC environment. For example, if your Design object is named "90123400.ASM" you may have a mapping defined to link

this Design to a Part named “90123400” (as shown above).

Alternatively, the Part number may be defined as a CAD property value, or simply a PLM autonumber. In any case, if the target Part number already exists, then the Assign function will simply link the Design to this existing Part. If it does not exist, it will create it. The linking operation is accomplished using the Agile PLM *Relationships* tab, and the link applies across all Design versions and Part revisions. It is simply expressing that *this Part is related to that Design*.

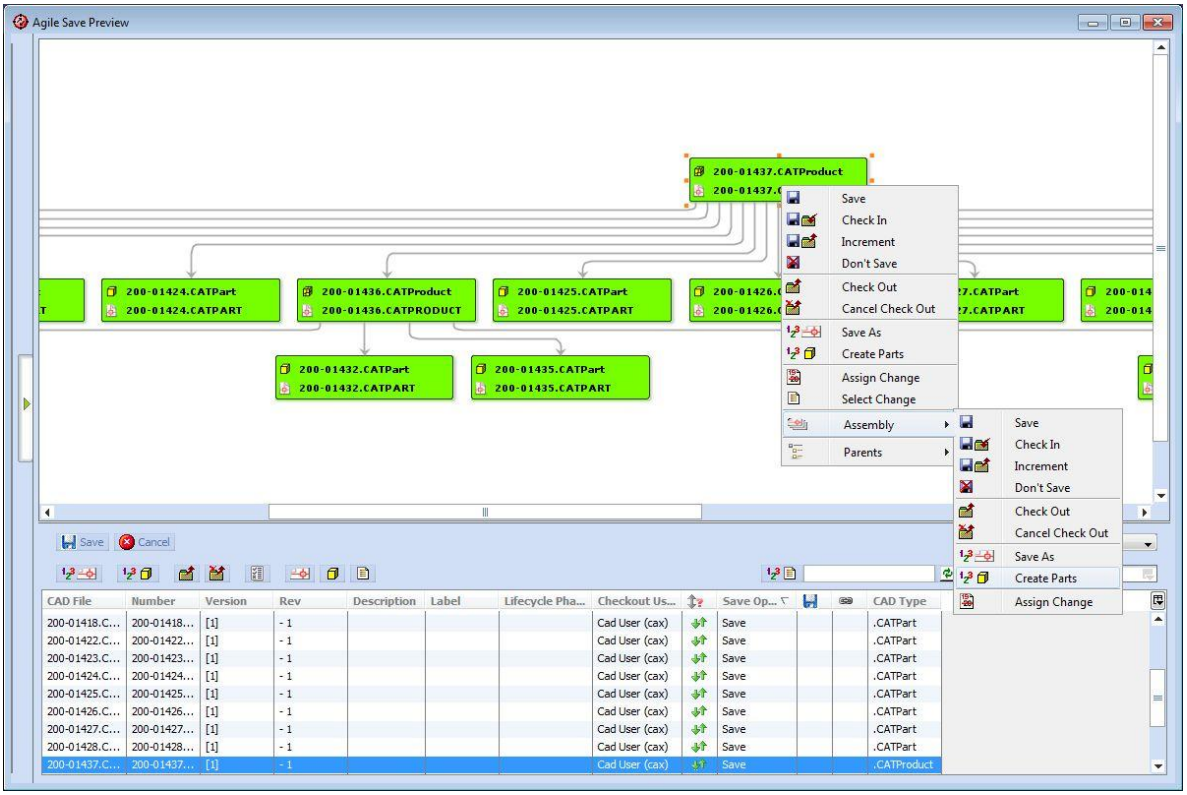
**Publish BOM** – This step actually creates or updates the BOM structure for all the assigned Items, based on the corresponding Design structure. Additionally, it attaches specific Design file types to the Item, as configured by your administrator.

**Manual BOM Edit** – This step does not occur within the EC Web Connector. If necessary, manual BOM edits can be made using the normal Redline BOM capability in Agile PLM. Manual BOM edits are tracked independently and will not be changed upon subsequent BOM publish updates.

# Using the Save Command with Publish Options

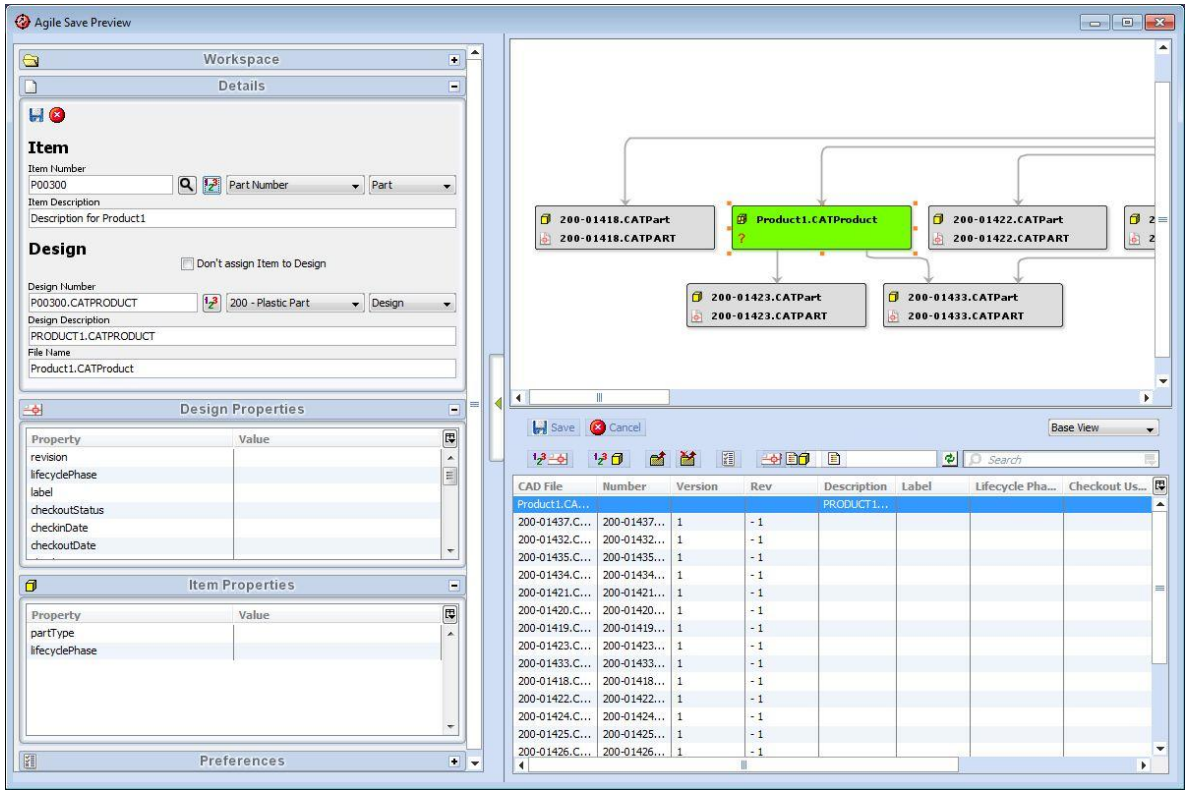
When you execute the *Agile → Save* command, the EC Web Connector pops forward, and displays a dialog similar to the Save dialog, as shown in the figure below. The *Item assignments* view inside the Save Preview displays the detailed information, which items are assigned, and which change order is active.

**Figure: Save Dialog**




From this starting point the user can execute the Assign function for all Designs using the *Assembly → Create Parts* button on the top level object, or for specific selected Designs using the *Create Parts* command on the context menu. During the assignment process, the user has the ability to interactively input Item attributes by double-clicking a node or row in the table (see figure below). After a Design object has been assigned, an icon appears in the *Assigned* column, and the Item number and other attribute information appear in the dialog.

**Figure: Assign Process**

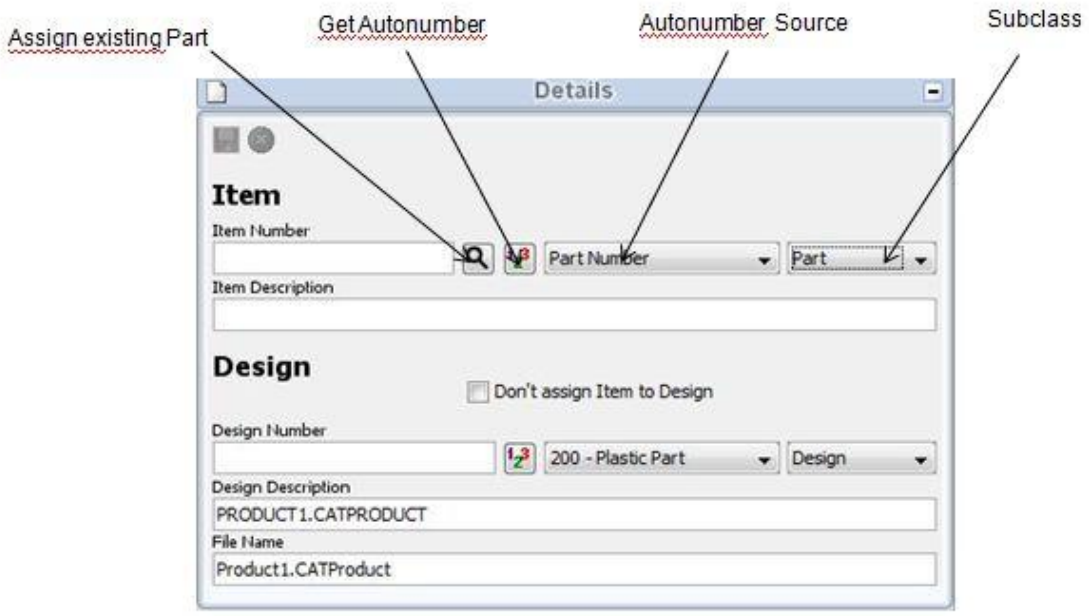



Once the assign process is finished, the user can use the *Save* button to complete the process. The Assign process can be automated during Save As, if the *Save As* preference is set to *Part and Design* and the *Part Creation* preference is set to *Create and Link*. Then the manual assignment step can be overridden.

Please refer to the *Save Section* in this manual for more information about the context menus and displayed tables.

Details of the interactive Create Part dialog are shown below.

**Figure: Interactive Create Part Dialog**




| | |
|---|---|
| Number | The value that will become the Number assigned to the Agile Part that is being created. |
| Description | The value that will become the Description assigned to the Agile Part that is being created. |
| Filename | The CAD filename belonging to the Design, related to the Part being created. |
| Autonumber Button | If you click the button, it will put the next available autonumber from the selected sub-class and autonumber, into the Number field. |
| Part sub-class | Selector that allows you to pick the Part sub-class to use for creating this Part in Agile. This is for overriding the default value set in the main Save dialog. |
| Part Autonumber | Selector for the autonumber to use for creating this Part in Agile. |
| Property / Value Area | This area displays, and allows editing, for other properties that are being set from CAD into Agile. Properties can be either text or list values. Text values are simply typed in, while list values are selected from a list. |

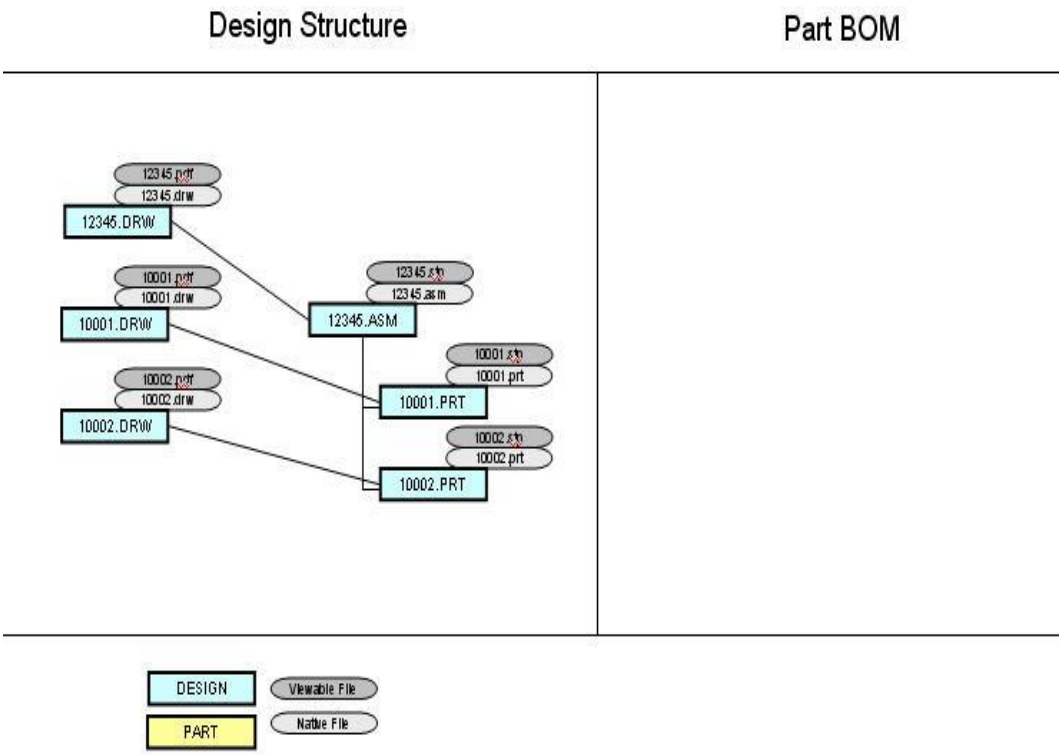
**Note** Your site most likely has pre-defined mappings for Number, Description, and other properties. You should check with your administrator to understand the allowable values to use. Also, these properties can be set as “Required”, meaning that you must enter a value before exiting the dialog.

# Details of the BOM Publishing Process

The following diagrams explain in detail how the data structures in PLM evolve as you go through the Save process with BOM Publishing enabled. It is important to understand the difference between 3D and 2D Designs, and how they are used to publish the BOM.

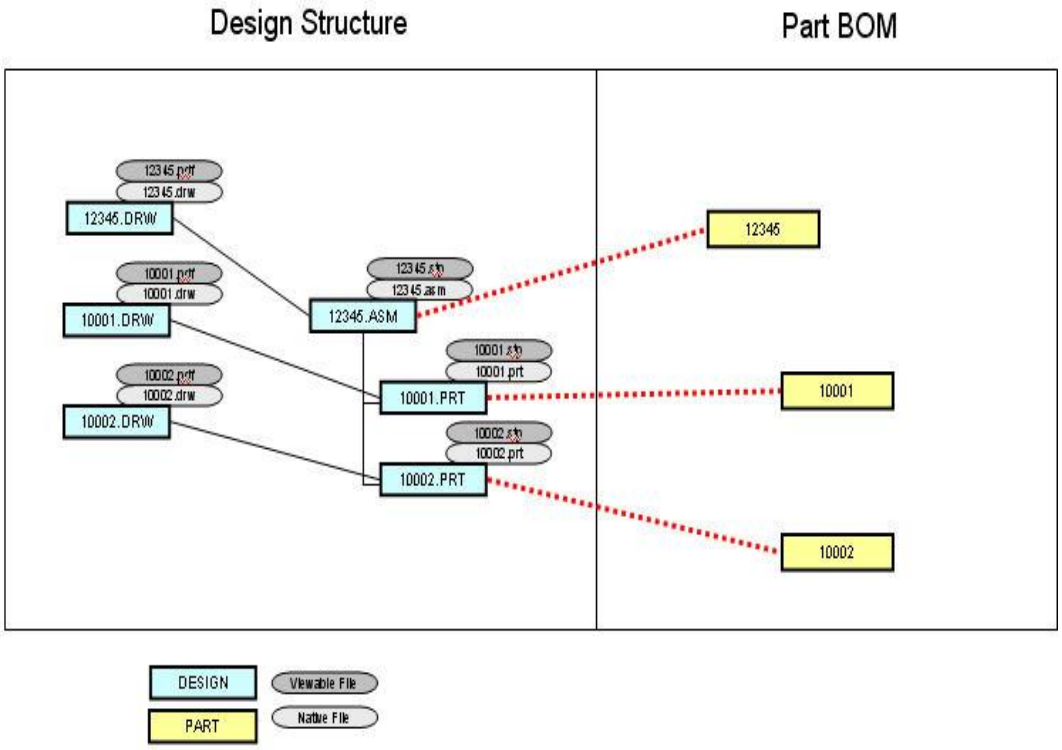
As a starting point, this diagram shows a typical Design structure containing both 3D and 2D files. Note that the 2D drawing objects are stored as parent objects of the corresponding 3D object, and that both 3D and 2D Designs can contain native and viewable file types.

**Figure: Details of BOM Publishing Process – Step 1**



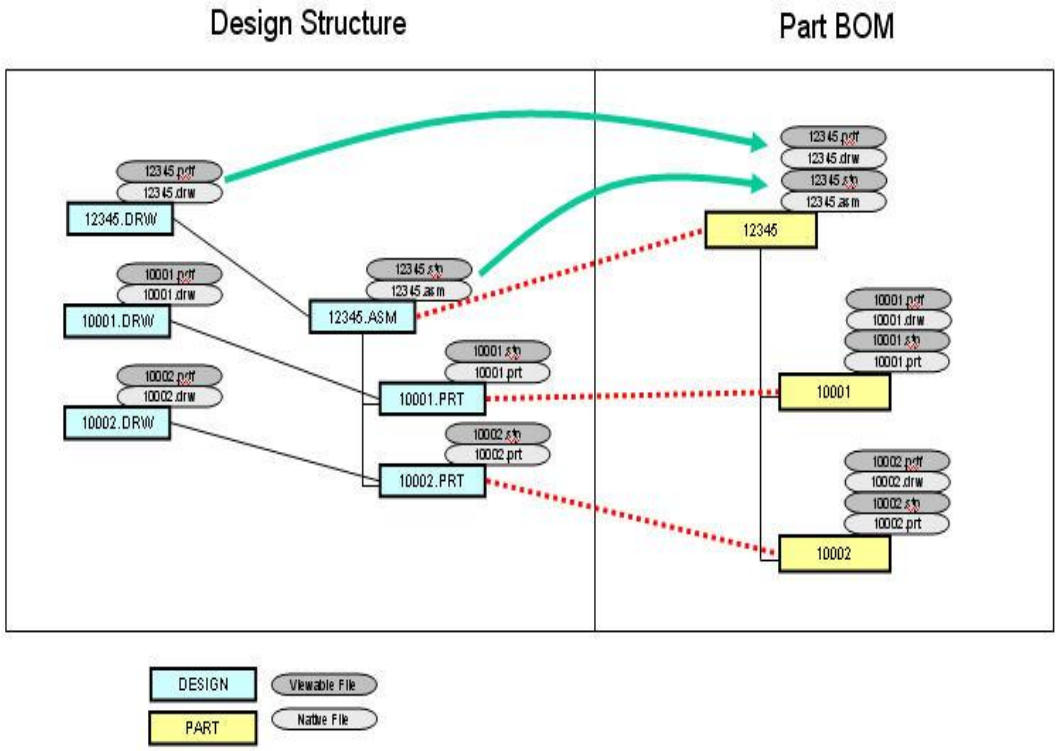

The next step is the assignment between the Design and the Part. It is important to recognize that the structure driving the BOM Publishing process on the Design side is the **3D structure**. This is because the 3D structure mimics in CAD the actual physical product. After the assignment process, the 3D Design is linked to the corresponding Part by a PLM Relationship. Note that at this time there is no BOM structure between the Parts (unless it existed already).

**Figure: Details of BOM Publishing Process – Step 2**




After the assignment, the next step is to publish. Not only does this create or update the BOM structure between the Parts, but it can also attach files from the Design objects to the Parts, depending on your preference settings. Note that files from the assigned 3D Design plus those from that object's 2D Designs can be attached. This provides a powerful method to collect the necessary CAD files onto the Part objects that will be viewed throughout the organization.

**Figure: Details of BOM Publishing Process – Step 3**




# Configurations and Family Tables

Pro/E has special capabilities for dealing with families of parts. In each case, the EC Web Connector will create correct BOM structures. Family table generics and instances are both maintained within the design structure. When using the publishing options within the Save command, a unique part will be generated for each instance or generic that is directly referenced within an Assembly. No special parameter definition is required.

# Change Process For Parts

The Save command provides access to the ECO change process for Parts. While the process can be initiated from the EC Web Connector, most of the workflow takes place using the standard Agile Web Client.

# Property Mapping

## Introduction

Properties (also known as parameters, attributes or metadata) are information stored as text strings that are associated with CAD data. Examples are part number, description, and author. The EC Web Connector supports bi-directional transfer of properties between CAD and Agile. In other words, you can enter a property in CAD and have it be put into Agile, or vice versa. Properties are useful in the definition and classification of your design data, and are also useful for searching. The specific mapping of properties at your site is defined in the configuration file by your system administrator.

## Types of Mapping

Multiple types of mappings are supported by the CAD Connectors. When mapping from CAD to Agile, there are two types of properties that can be mapped – system properties and user properties. System properties are not directly defined by the user, they are things like the filename and the CAD software version number, which can be saved as properties in Agile. User Properties are defined by the user with the following commands:

Pro/E: *Tools → Parameters*

CATIA V5: *Tools → Formula*

Property mapping supports the following types of Agile attributes:

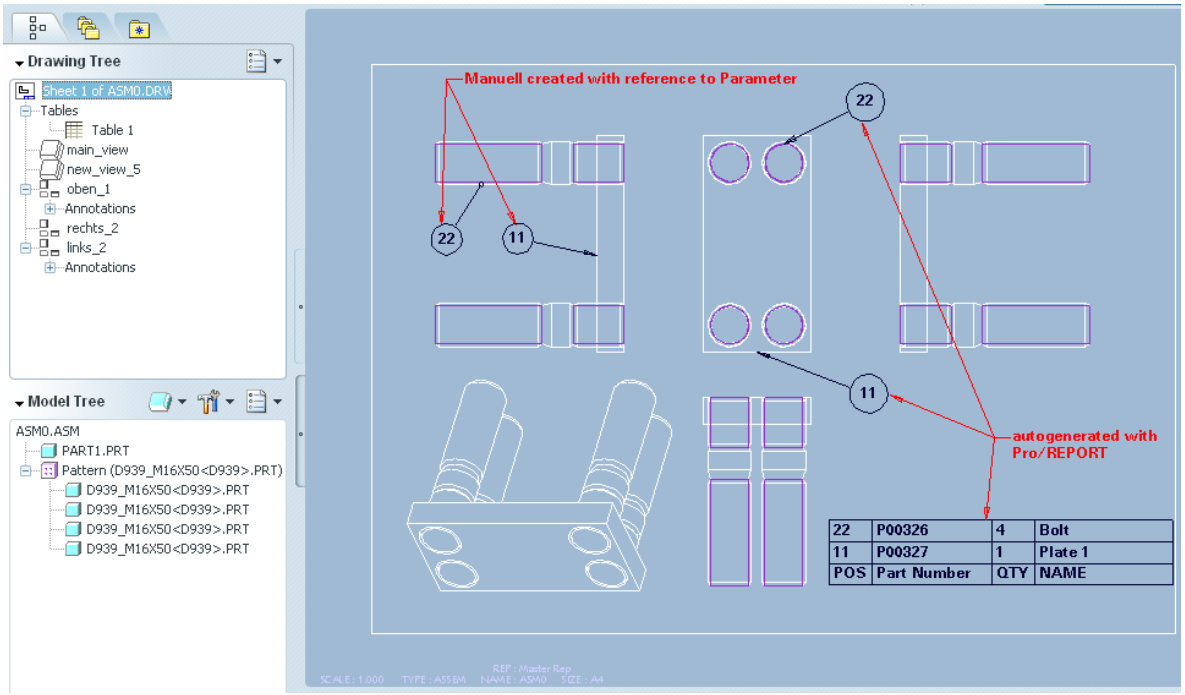
Text, MultiText, List, and MultiList.

System properties and user properties are mapped into Agile Designs and Parts as part of the *Save* command.

Mapping from Agile into CAD is done using the *Update Properties* command. It can also be configured to occur automatically during the *Save* process. Properties from both the CAD Design and the associated Part object can be mapped into CAD.

When working with drawings, there is another available command called *Update Title Block*. This updates properties just for the current drawing, not all subordinate models. In order to use properties within a Title Block, you need to define the text notes to be linked to properties, either within the drawing or within the part or assembly referenced on the drawing. This is standard CAD functionality. Figure below shows an example of properties used in notes within a Title Block.

**Figure: Properties used in Title Block**



# CAD specific Functionality

Some CAD Connector functionality is specific to certain connectors, because of unique capabilities in certain CAD tools. This section provides details on those specific functions.

## Handling of Part Families and Configurations

Companies working with Pro/E or Solid Works use configurations and part families. Especially in Pro/E and in Solid Works there is no real file for configurations. The configurations can only change if the master file (Generic) is touched and modified. The selection logic inside the EC dialogs is like this:

The selection between the generic and the instances or configurations is transparent.

- If a Generic is **selected**, all instances in session are also **selected**.
- If a Generic is **deselected**, all instances are also **deselected**.
- If an Instance is **selected**, the related Generic is also **selected**.
- If an Instance is **deselected**, the related Generic is **NOT deselected**.

## Numbering of Part Families and Configurations

By default, the EC connectors treat configurations as regular files, so each configuration gets a unique number assigned in PLM. The numbers do not have to match between the generic and the instances from the PLM point of view.

Optionally, number of instances can derive from the generic number by adding a suffix. EC offers two options controlled by the switch `FamilyInstanceNumbering` in **CAXConfig.xml**:

- `GENERIC_INDEX` – appends a counter to the generic number

- `GENERIC_CONFIG` – appends the configuration name to the generic number

The resulting instance number is cut to 50 chars and any special characters are removed.

# Family Table Handling – Pro/ENGINEER

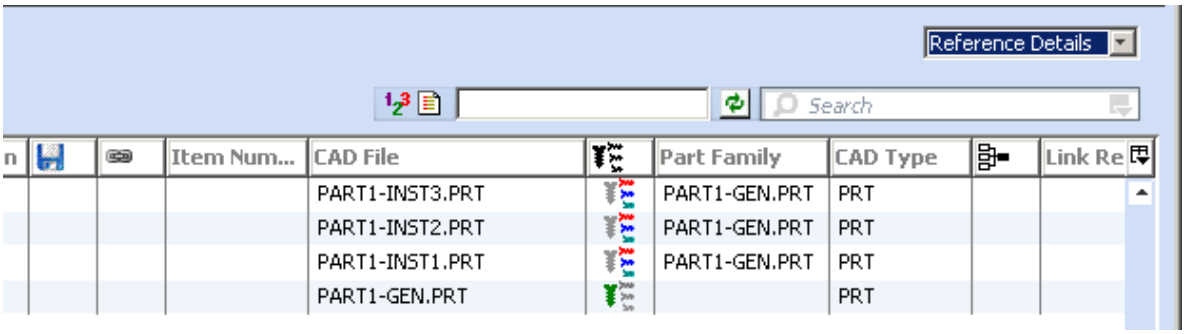
## Introduction

Pro/ENGINEER contains specialized functionality to manage families of parts and assemblies. This is called "Family Tables" in Pro/ENGINEER. The Agile CAD Connectors for these two tools provide additional functionality within the EC Web Connector to display and manage part family information.

## EC Web Connector User Interface (Design)

In all main EC Web Connector dialogs – Save and LoadPreview, additional columns display the part family information in the *Reference Details View*. See Figure 3-3 for an example showing the Save dialog. If no family table parts are contained within the current CAD model, then the extra columns are not even displayed.

**Figure: Part Family Columns**




The additional columns are:

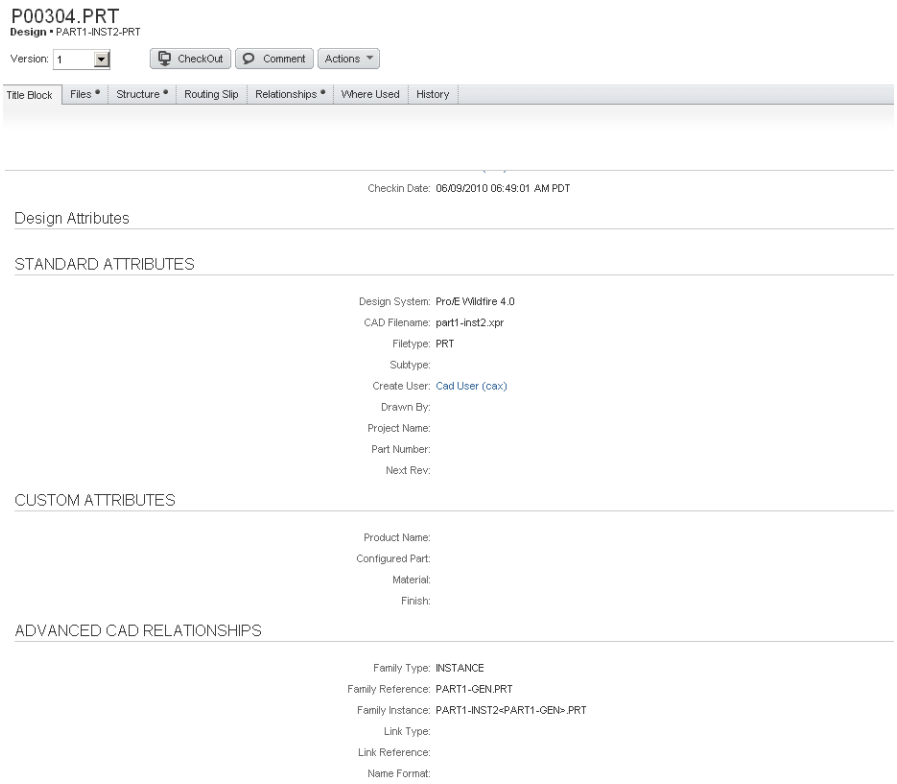
- *Model Type* – An icon column, where the icon indicates either an instance or a generic.
- *Part Family* – Lists filename of the Generic model.

A Generic is indicated by the *Generic* icon in the *Model Type* column, and nothing in the *Family Reference* column.

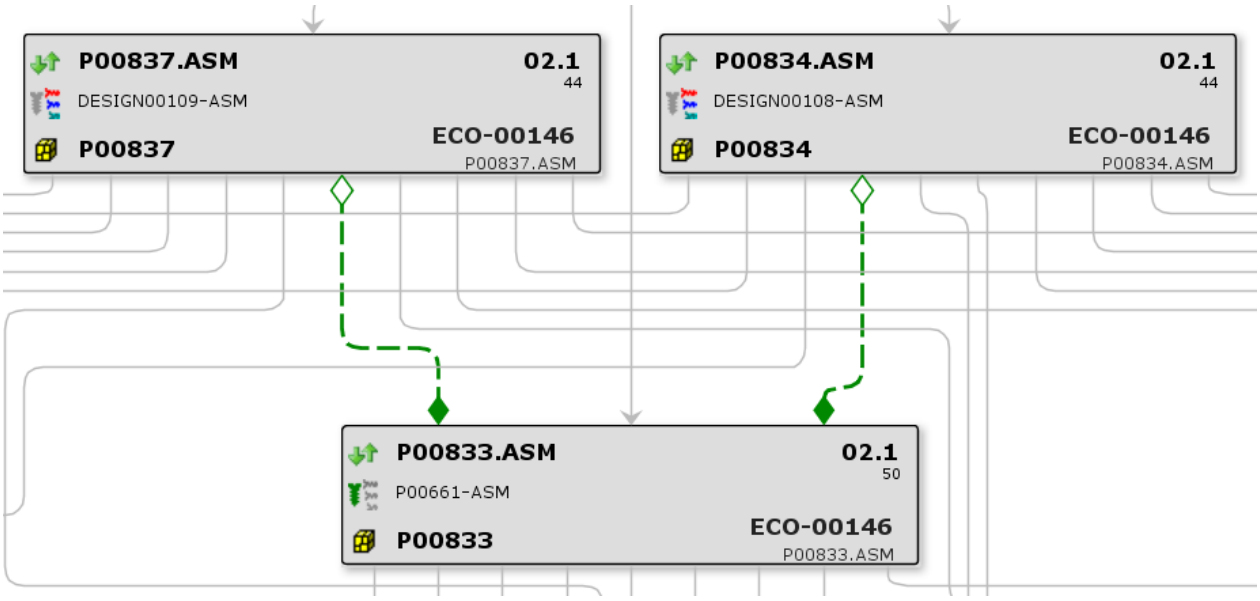
An Instance is indicated by the *Instance* icon in the *Model Type* column, and the Design number and filename of its corresponding Generic in the *Family Reference* column.

The information shown in these columns will also be shown on the form of each object in the EC Web Connector, such as the *CAD Model Type* and *CAD Model Reference* shown in figure below (note that this form will look somewhat different in each customer environment). You can use these attributes to perform searches, for example, on instances or generics.

**Figure: CAD Model Type - CAD Model Reference**



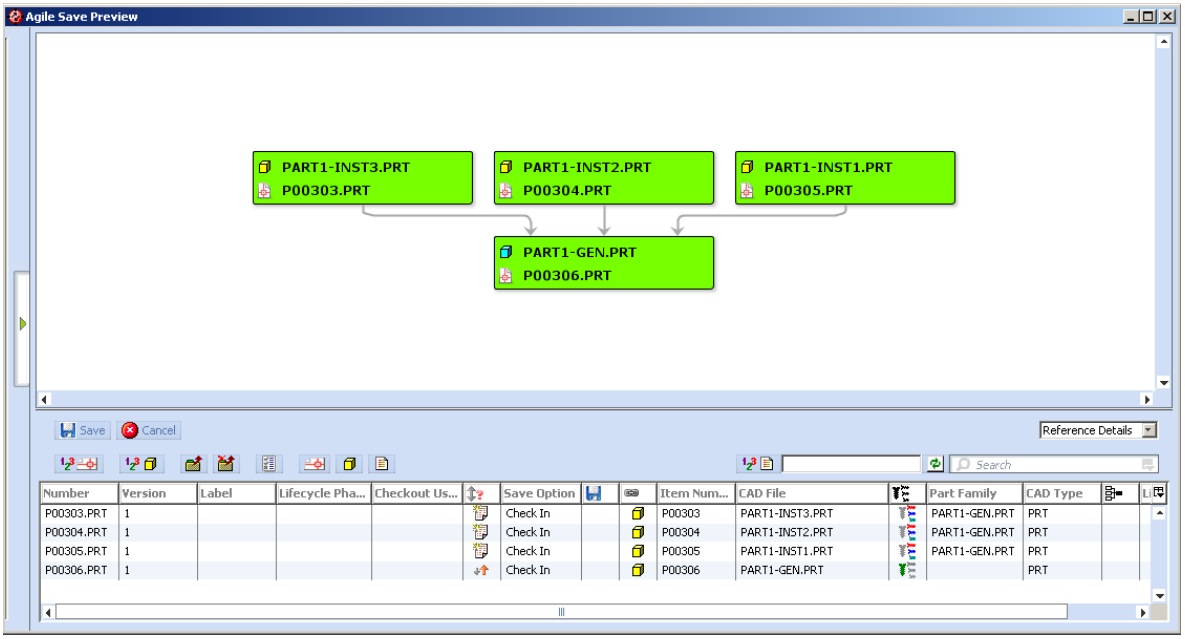

The relationship between Generic and Instances is highlighted in the browser view with green dashed lines and special end nodes. Besides this, the Part Family indicators are displayed in the nodes and in the References list view.




## "Save Family Table" Command

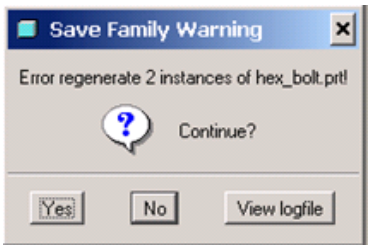
In the Agile menu within the CAD tool, there is a command called Save Family Table, which allows you to save an entire part family at once into Agile. It brings up the Save dialog containing the generic and all instances of a part family. In order to use this function, you must have a part family generic part or assembly active in your CAD session. This command allows the user to save or update all instances of a part family, including the generic all at once.

**Figure: Save Family Table**



When using Pro/ENGINEER, this command also validates each instance and prompts you if there is an error with any of the instances. If there are any errors, the following dialog appears, and allows you to view the log file of errors (which is also stored in your log directory for further access). If you choose to continue, the contents of the Save dialog will contain only the properly validated instances.

**Figure: Save Family Warning**



# External Reference Handling – Pro/ENGINEER

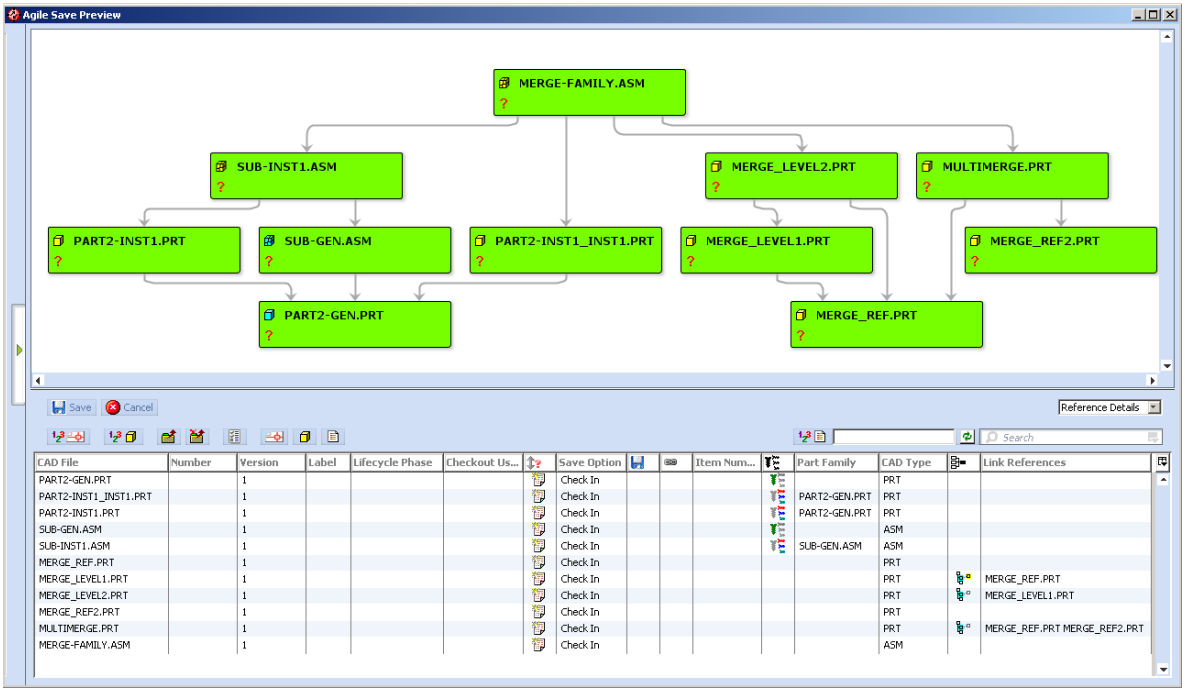
## Introduction

Pro/ENGINEER contains specialized functionality to manage external references such as copy geometry, publish geometry, merge/inheritance and shrinkwrap. The Agile CAD Connectors provide additional functionality within the EC Web Connector to display and manage external reference information.

## EC Web Connector User Interface

In the EC Web Connector dialogs – Save and Load Preview, additional columns display in the *Reference Details View*. See the example below for an example showing the Save preview. If no external references are contained within the current CAD model, then the extra columns will be empty.

**Figure: External References (Save Preview)**




The columns are:

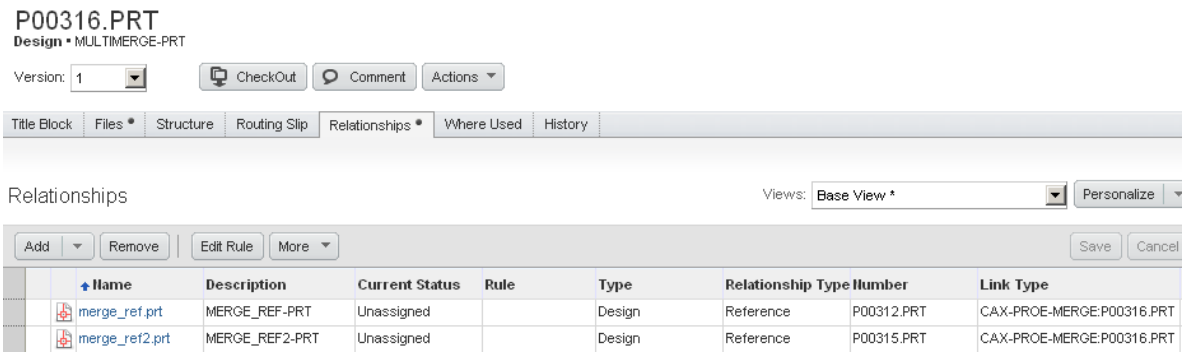
- Link Type – An icon column, the icon indicates that the file either contains reference geometry or is the referenced geometry.

- Link References – Lists the name(s) of the referenced model(s).

The information shown in these columns is also shown on the form of each object in the EC Web Connector. You can use these attributes to perform searches, for example on merge reference parts.

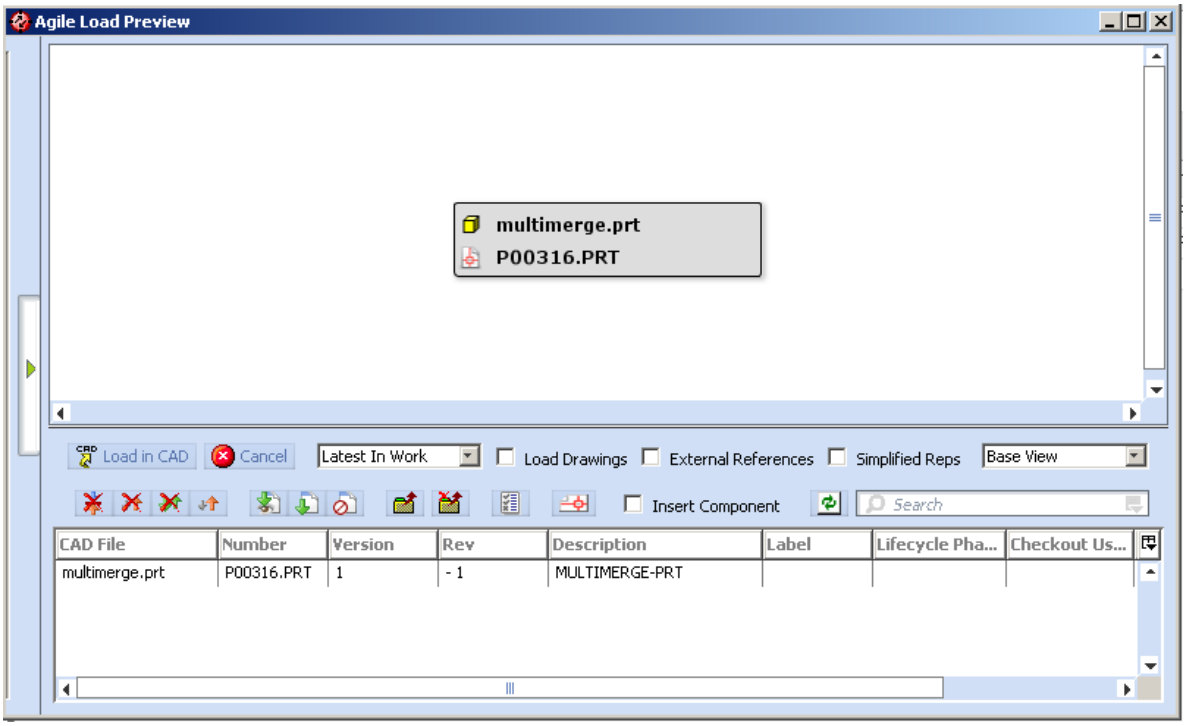
The external references are stored as additional relations on the Relationships tab in Agile.

**Figure: Relationships tab in Agile**




External references stored in the Relationships tab in Agile are supported within the Load process.

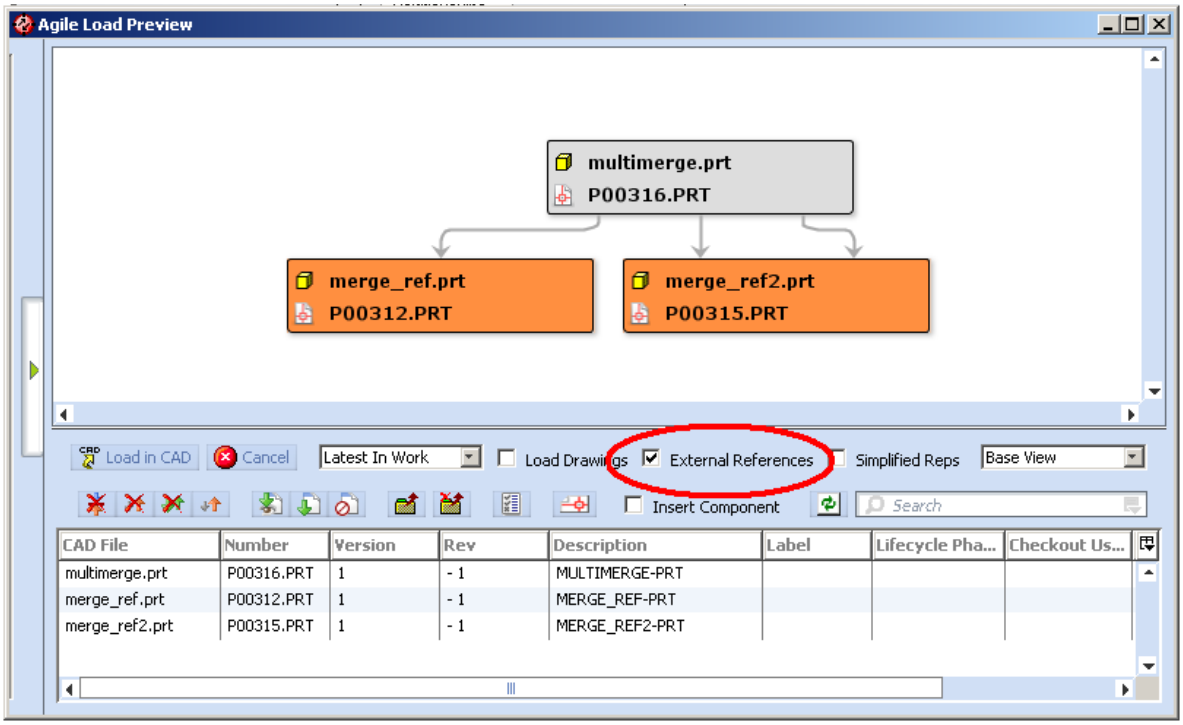
**Figure: Agile Load Preview without External References**




If you check the *External References* box in the *Agile Load Preview*, additional references are

displayed in the preview.

**Figure: Agile Load Preview with External References**



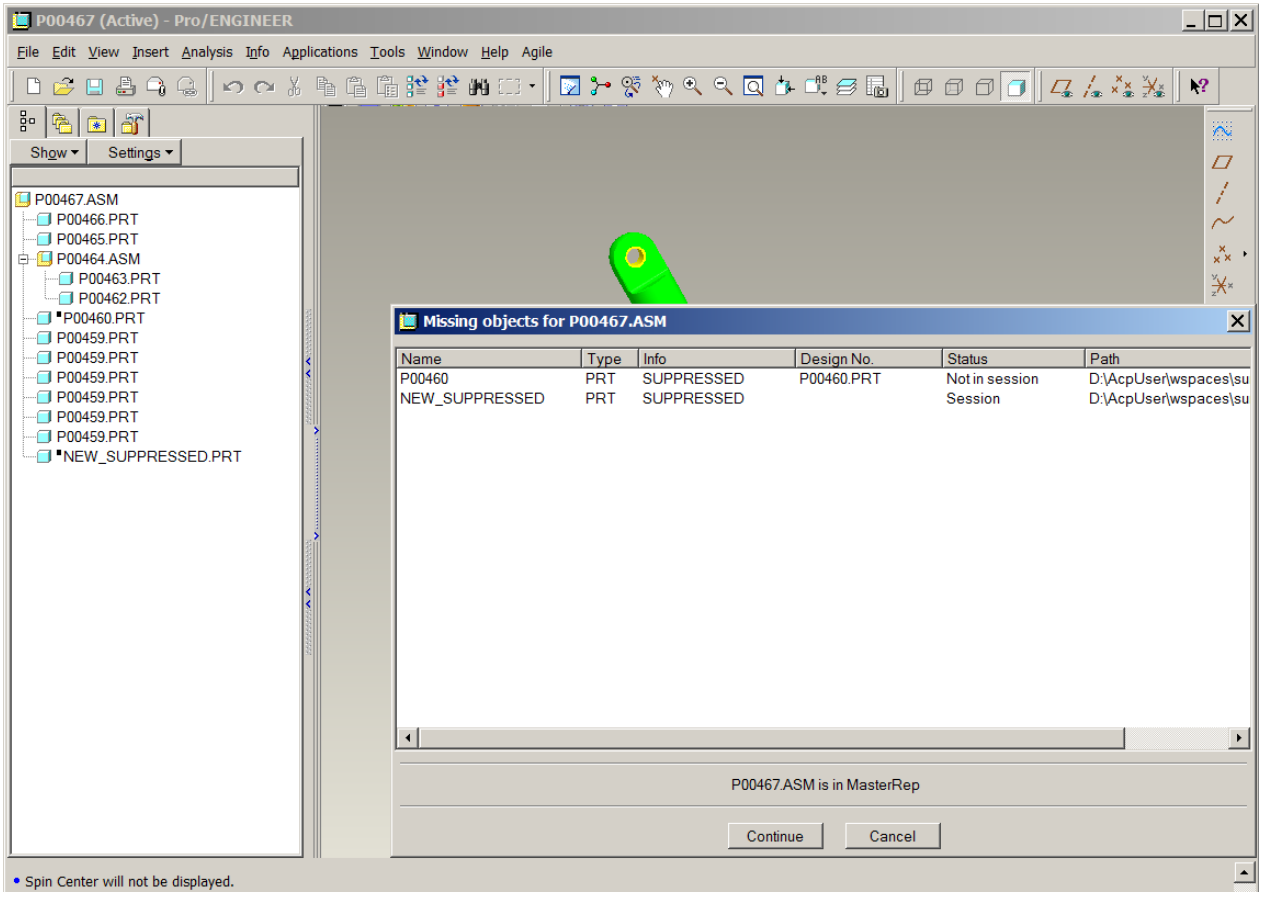

Load commands will checkout the additional files. For automated loading the reference files in the Pro/ENGINEER session you have to set the two config.pro settings:

RETRIEVE_MERGE_REF_PARTS YES

RETRIEVE_DATA_SHARING_REF_PARTS YES

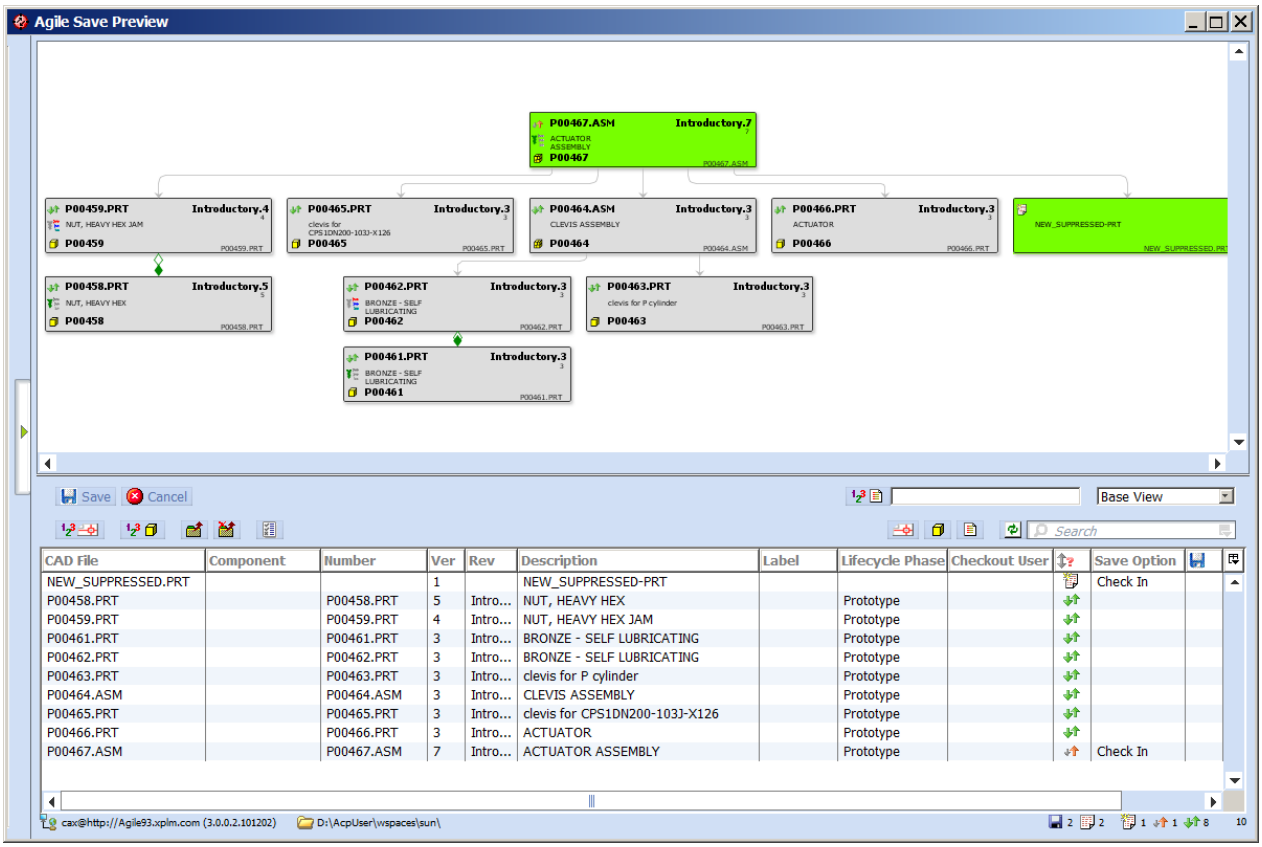
# Support for Suppressed Components – Pro/ENGINEER

The Pro/ENGINEER connector is extended to handle suppressed components when saving. Before saving the connector scans the structure of the current object and detects suppressed components. If suppressed components are found, an information dialog is shown.

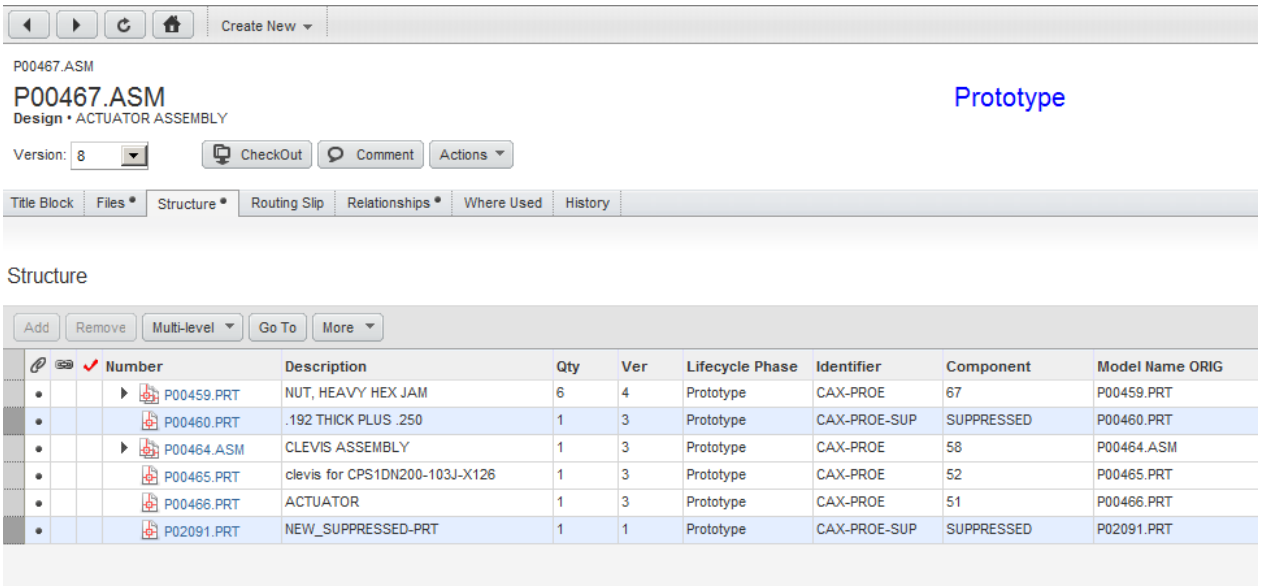



If a suppressed component is found in PLM, it will not be shown in the Save Preview. If a suppressed component is not found in PLM and it is in session, it will be shown in the Save Preview and handled like an active component.

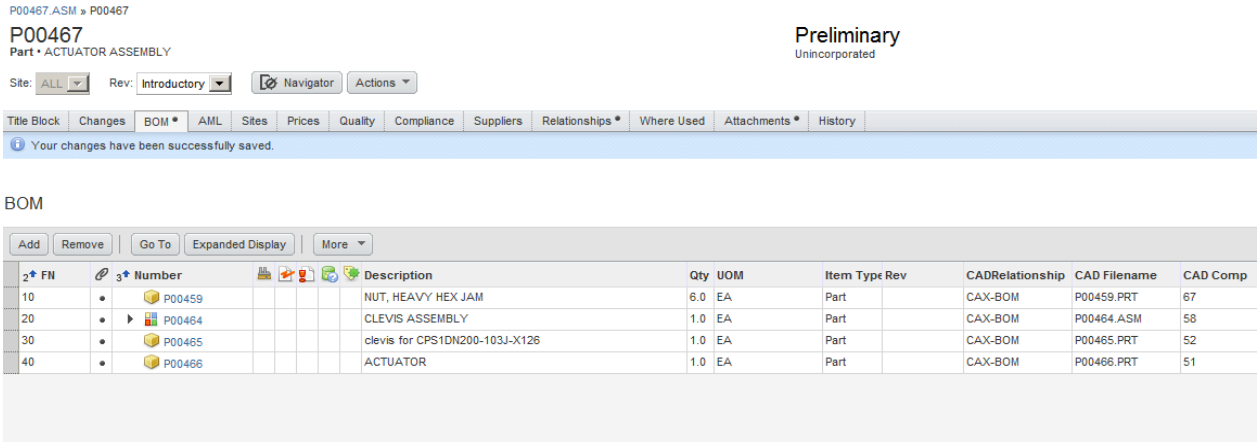
**ATTENTION: Components not found in PLM and not in session, cannot be handled!**




Suppressed components with a PLM Design number are built into the Design structure with a special identifier “CAX-PROE-SUP” and component information “SUPPRESSED”.

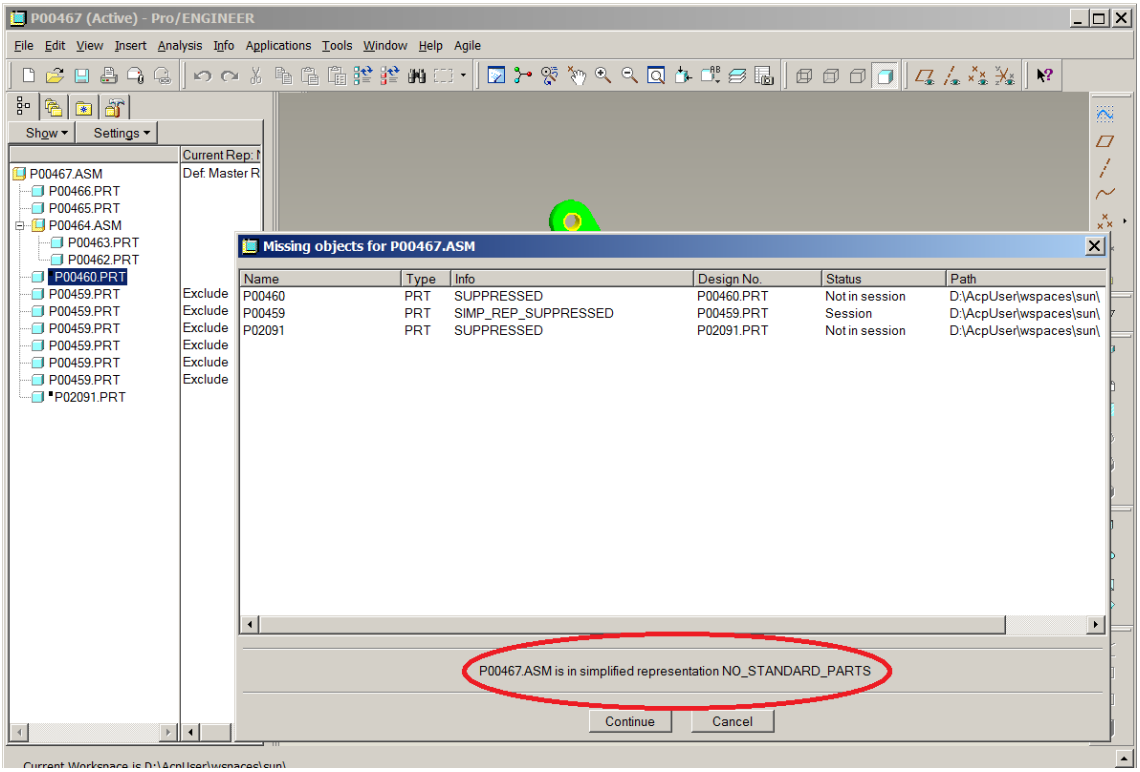



Suppressed components are NOT BOM relevant and therefore filtered from the Item BOM during publish.



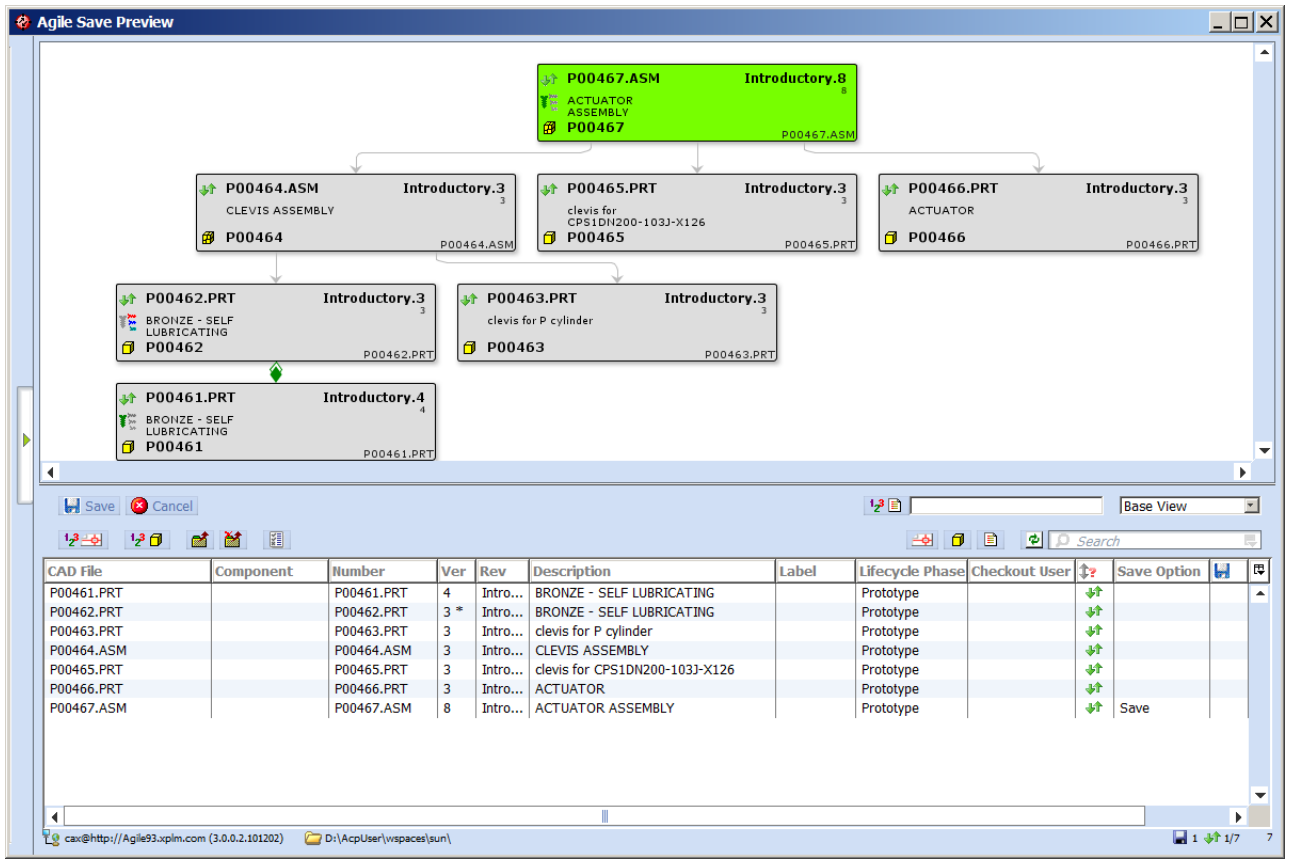

# Simplified Representations – Pro/ENGINEER

The handling of components suppressed by a simplified representation is similar to the handling of suppressed components. In addition, the information dialog shows the current simplified representation.

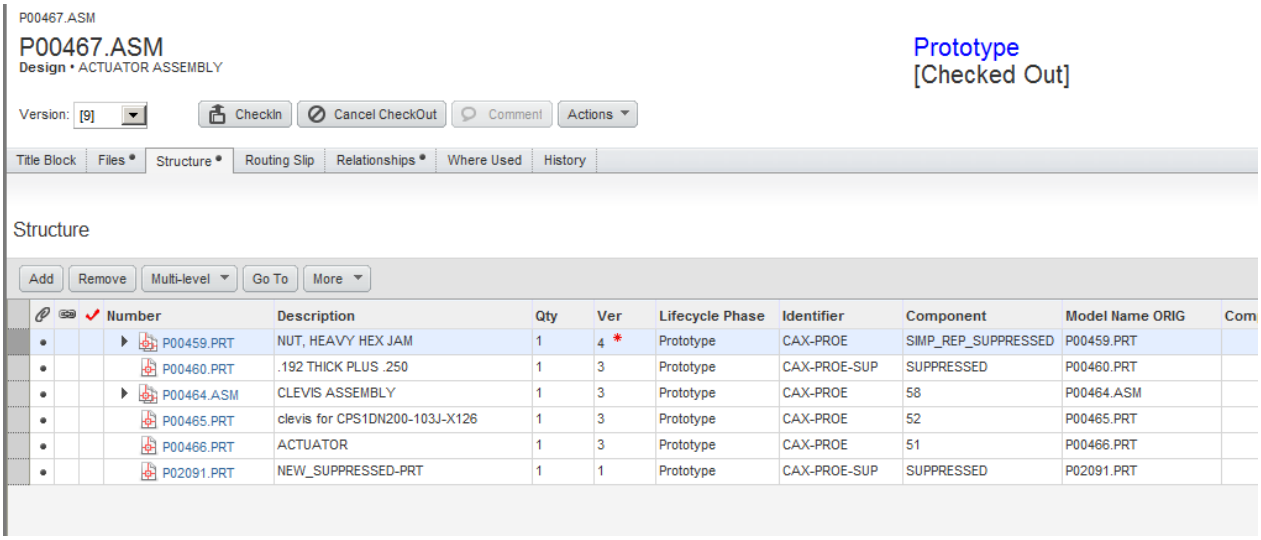



If a component suppressed by a simplified representation is found in PLM, it will not be shown in the Save Preview. If a component suppressed by a simplified representation is not found in PLM and it is in session, it will be shown in the Save Preview and handled like an active component.

**ATTENTION: Components suppressed by a simplified representation not found in PLM and not in session, cannot be handled!**

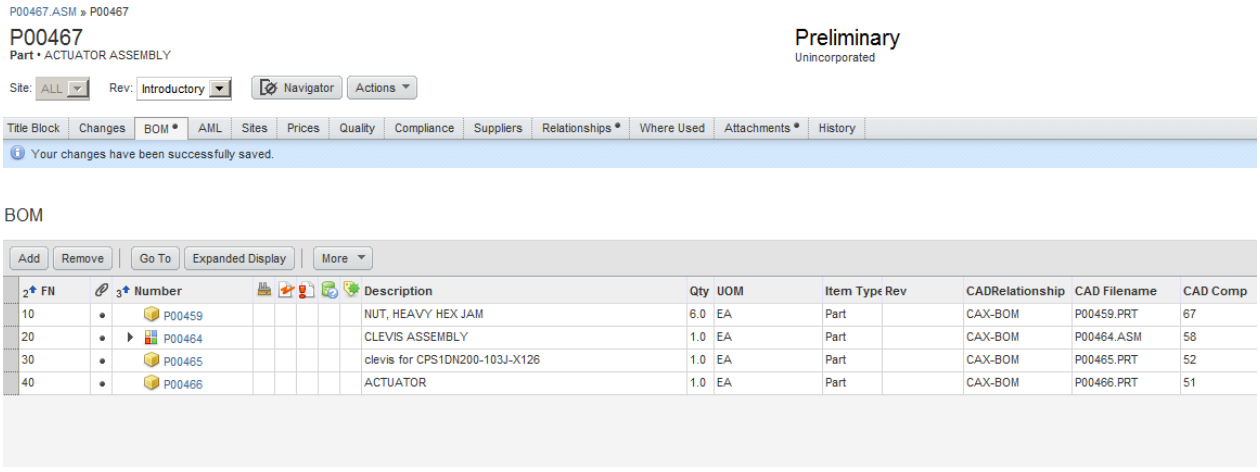

Agile Save Preview
P00467.ASM Introductory.8
ACTUATOR ASSEMBLY
P00467
P00467.ASM
P00464.ASM Introductory.3
CLEVIS ASSEMBLY
P00464
P00464.ASM
P00465.PRT Introductory.3
clevis for CPS1DN200-103J-X126
P00465
P00465.PRT
P00466.PRT Introductory.3
ACTUATOR
P00466
P00466.PRT
P00462.PRT Introductory.3
BRONZE - SELF LUBRICATING
P00462
P00462.PRT
P00463.PRT Introductory.3
clevis for P cylinder
P00463
P00463.PRT
P00461.PRT Introductory.4
BRONZE - SELF LUBRICATING
P00461
P00461.PRT
Save
Cancel
Base View
Search
CAD File | Component | Number | Ver | Rev | Description | Label | Lifecycle Phase | Checkout User | Save Option
P00461.PRT | | P00461.PRT | 4 | Intro... | BRONZE - SELF LUBRICATING | | Prototype | | |
P00462.PRT | | P00462.PRT | 3 * | Intro... | BRONZE - SELF LUBRICATING | | Prototype | | |
P00463.PRT | | P00463.PRT | 3 | Intro... | clevis for P cylinder | | Prototype | | |
P00464.ASM | | P00464.ASM | 3 | Intro... | CLEVIS ASSEMBLY | | Prototype | | |
P00465.PRT | | P00465.PRT | 3 | Intro... | clevis for CPS1DN200-103J-X126 | | Prototype | | |
P00466.PRT | | P00466.PRT | 3 | Intro... | ACTUATOR | | Prototype | | |
P00467.ASM | | P00467.ASM | 8 | Intro... | ACTUATOR ASSEMBLY | | Prototype | | Save
cax@http://Agile93.xplm.com (3.0.0.2.101202)
D:\AcpUser\wspaces\sun\
1
1/7
7

Components suppressed by a simplified representation with a PLM design number will be built into the Design structure like normal components (with identifier “CAX-PROE”) and component information “SIMP_REP_SUPPRESSED”.




| Number | Description | Qty | Ver | Lifecycle Phase | Identifier | Component | Model Name ORIG | Com |
|---|---|---|---|---|---|---|---|---|
| P00459.PRT | NUT, HEAVY HEX JAM | 1 | 4 * | Prototype | CAX-PROE | SIMP_REP_SUPPRESSED | P00459.PRT | |
| P00460.PRT | .192 THICK PLUS .250 | 1 | 3 | Prototype | CAX-PROE-SUP | SUPPRESSED | P00460.PRT | |
| P00464.ASM | CLEVIS ASSEMBLY | 1 | 3 | Prototype | CAX-PROE | 58 | P00464.ASM | |
| P00465.PRT | clevis for CPS1DN200-103J-X126 | 1 | 3 | Prototype | CAX-PROE | 52 | P00465.PRT | |
| P00466.PRT | ACTUATOR | 1 | 3 | Prototype | CAX-PROE | 51 | P00466.PRT | |
| P02091.PRT | NEW_SUPPRESSED-PRT | 1 | 1 | Prototype | CAX-PROE-SUP | SUPPRESSED | P02091.PRT | |

Components suppressed by a simplified representation are BOM relevant.

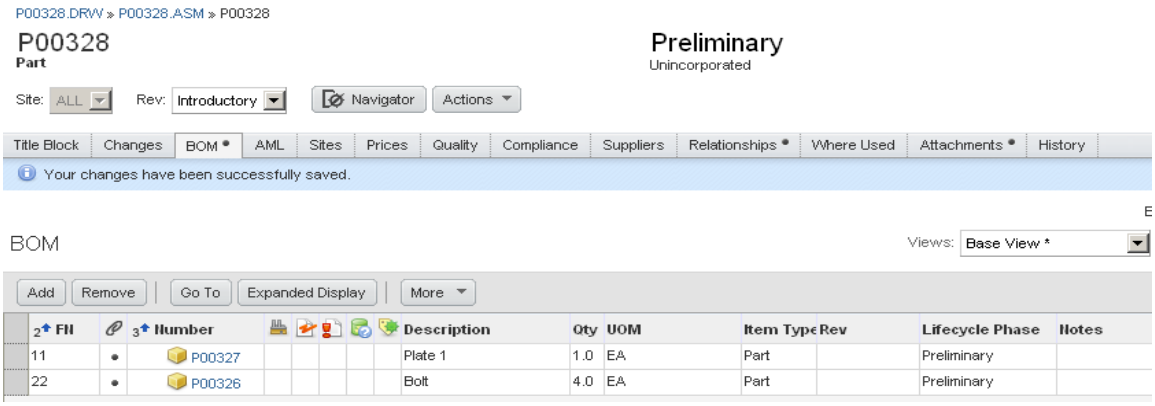



| FN | Number | Description | Qty | UOM | Item Type | Rev | CADRelationship | CAD Filename | CAD Comp |
|---|---|---|---|---|---|---|---|---|---|
| 10 | P00459 | NUT, HEAVY HEX JAM | 6.0 | EA | Part | | CAX-BOM | P00459.PRT | 67 |
| 20 | P00464 | CLEVIS ASSEMBLY | 1.0 | EA | Part | | CAX-BOM | P00464.ASM | 58 |
| 30 | P00465 | clevis for CPS1DN200-103J-X126 | 1.0 | EA | Part | | CAX-BOM | P00465.PRT | 52 |
| 40 | P00466 | ACTUATOR | 1.0 | EA | Part | | CAX-BOM | P00466.PRT | 51 |

# Using Agile Find Numbers – Pro/ENGINEER

Agile Find Numbers can be used in Pro/ENGINEER Drawings for BOM Balloons.

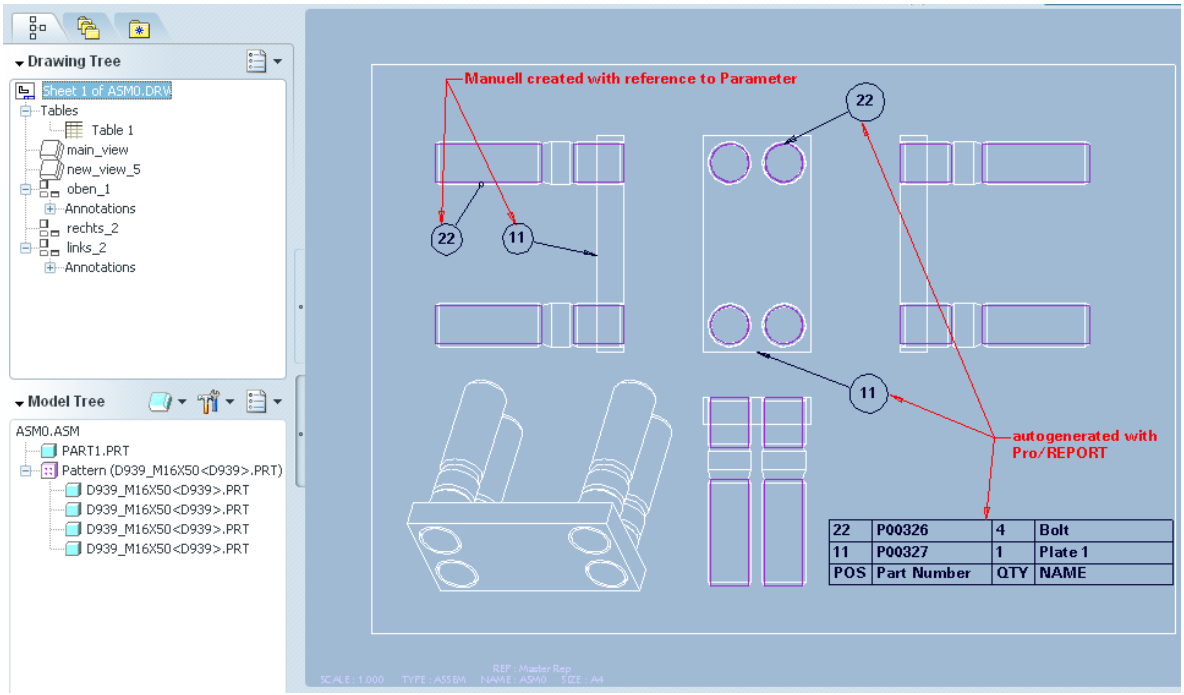
**Figure: Agile BOM with Find Numbers**



Find Numbers must be first input manually within the Part BOM in PLM. Once they are created, they can be accessed from Pro/E using either of these two options:

- Manually created balloons using the parameter "&AGILE_FIND_NO:att_cmp".
- Automated balloons using Report Parameter "asm.mbr.cparam.AGILE_FIND_NO" in a repeat region.

**Figure: Find Numbers in Pro/E Balloons**




# CGR File Handling – CATIA V5

## Introduction

Companies working with CATIA V5 commonly use CGR (CATIA Graphics Representation) files to simplify the representation of parts and assemblies that they are working with. The CGR format provides better performance when dealing with geometry that does not need to be modified (such as customer-provided assemblies that tooling is built from). The Agile CATIA V5 connector supports the management of CGR files.

## Functionality Overview – Datamodel

In keeping with Agile standard methodology, which is that all representations of a given CAD model are stored together in a common Design record, both the native CATPart or CATProduct file and the corresponding CGR file are stored in a common Design. Figure 1 shows the standard

CATProduct structure, while Figure 2 shows the same structure with an added CGR file.

**Figure: Standard CATProduct Structure in PLM**

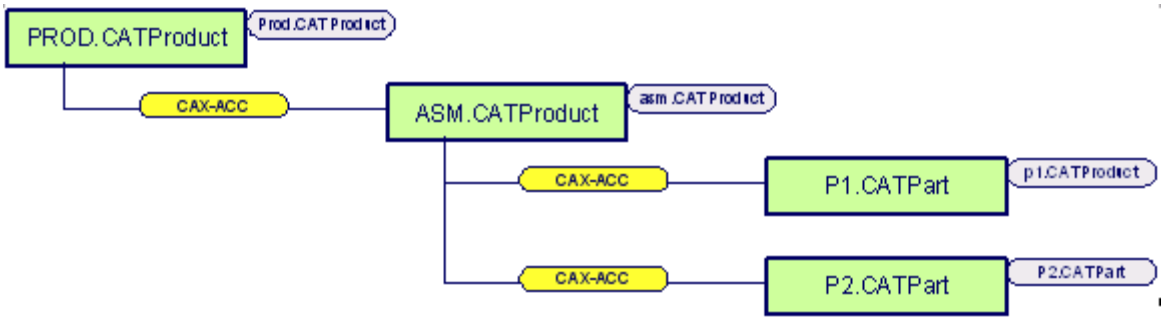
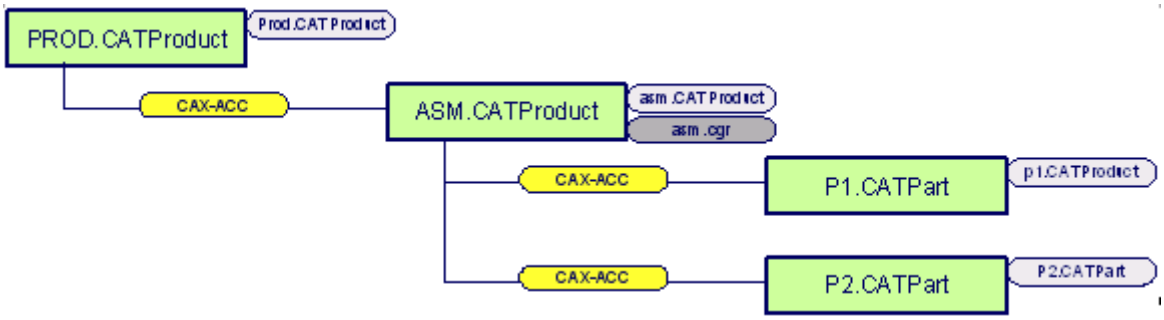



**Figure 3-16: Structure with both CATProduct and CGR in PLM**



The association between the CATProduct file and its corresponding CGR file is made through the new command *Save with CGR*. Once this has been done once, Agile knows that the two are associated. After that point, when the user uses the Load command to bring files from Agile into CATIA, the integration uses either the CGR or the native CATProduct, depending on whichever one was last saved (in its parent assembly) into Agile. This is “flagged” by a relationship attribute, denoted by the yellow bubble in figures 1 through 3. When the value is set to CAX-ACC by the Save command, upon subsequent loads the native CATProduct file will be loaded. When the value is set to CAX-ACC-CGR, subsequent Load commands load the CGR instead. Note that this flag is set automatically by the Save command and should not be set manually by the user, or data corruption may result.

**Figure: Assembly flagged to load CGR rather than CATProduct**

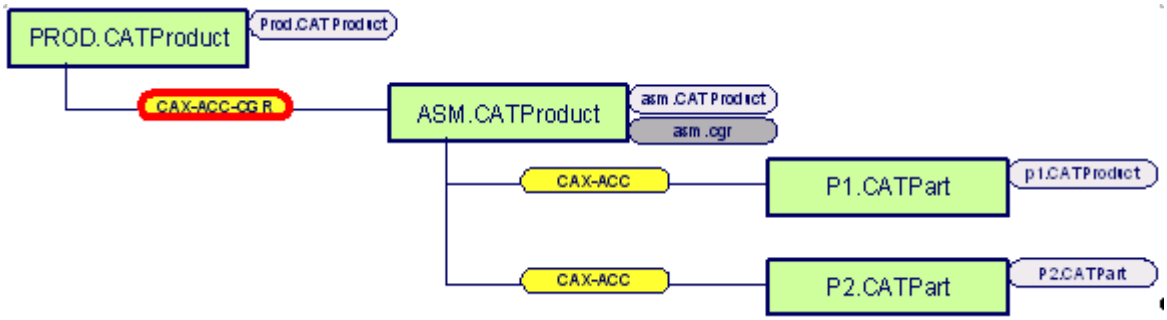
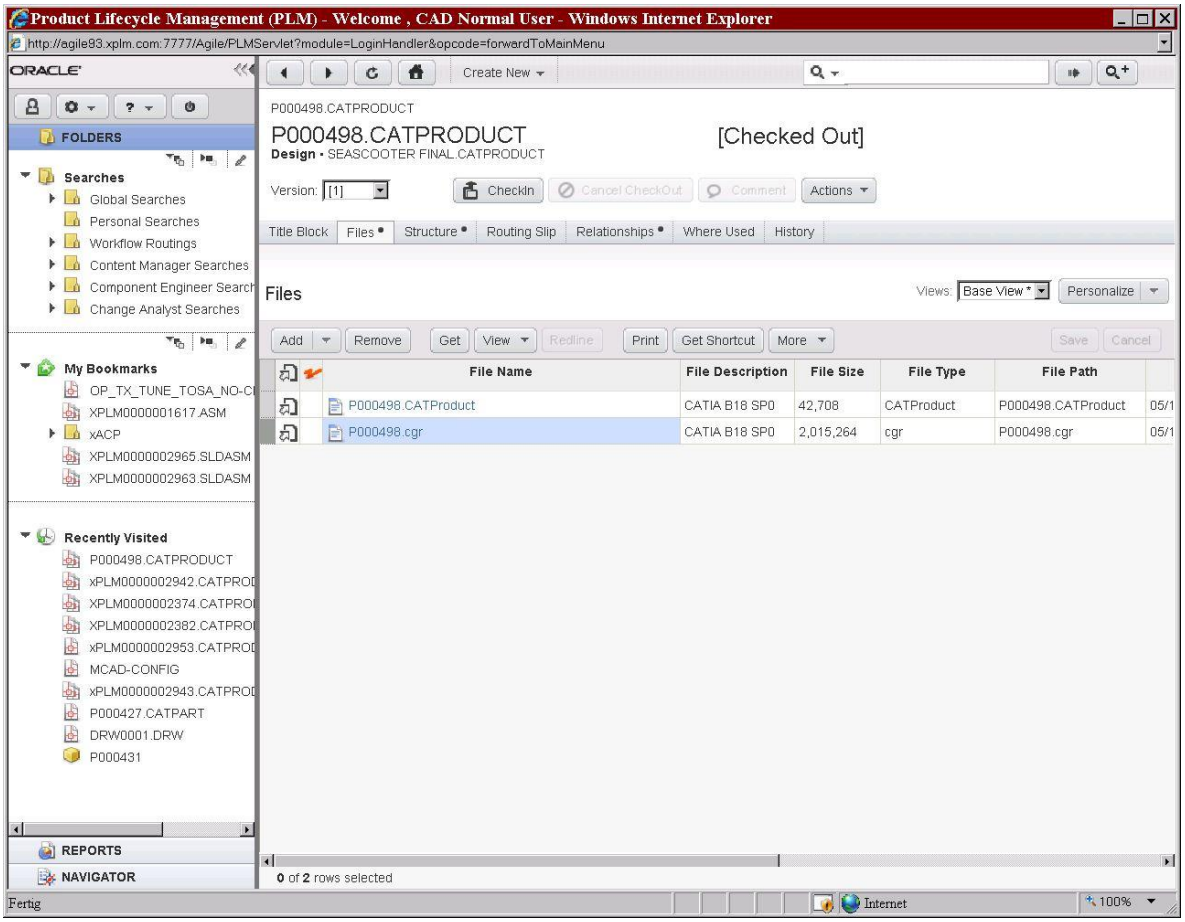


**Figure: Document containing CGR file**



The CATIA representation is shown in the next two figures:

**Figure: Screenshot from the Assembly PROD**

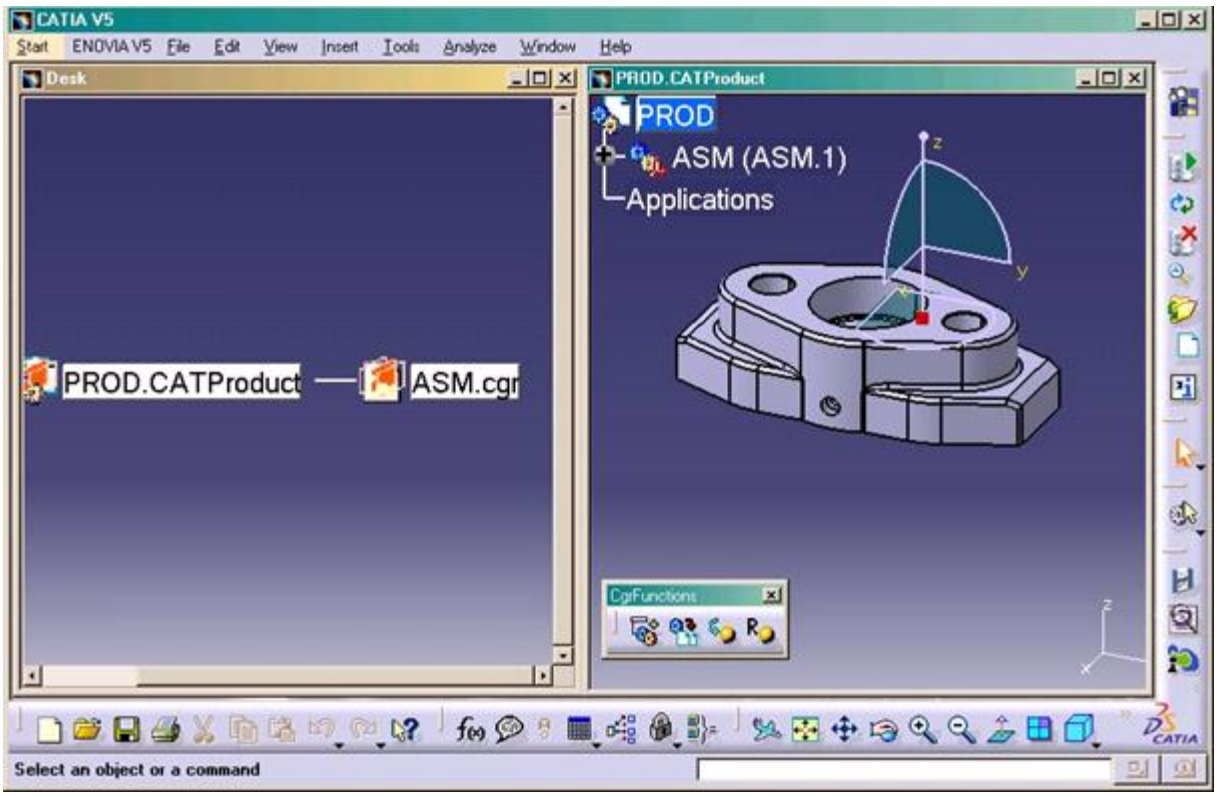
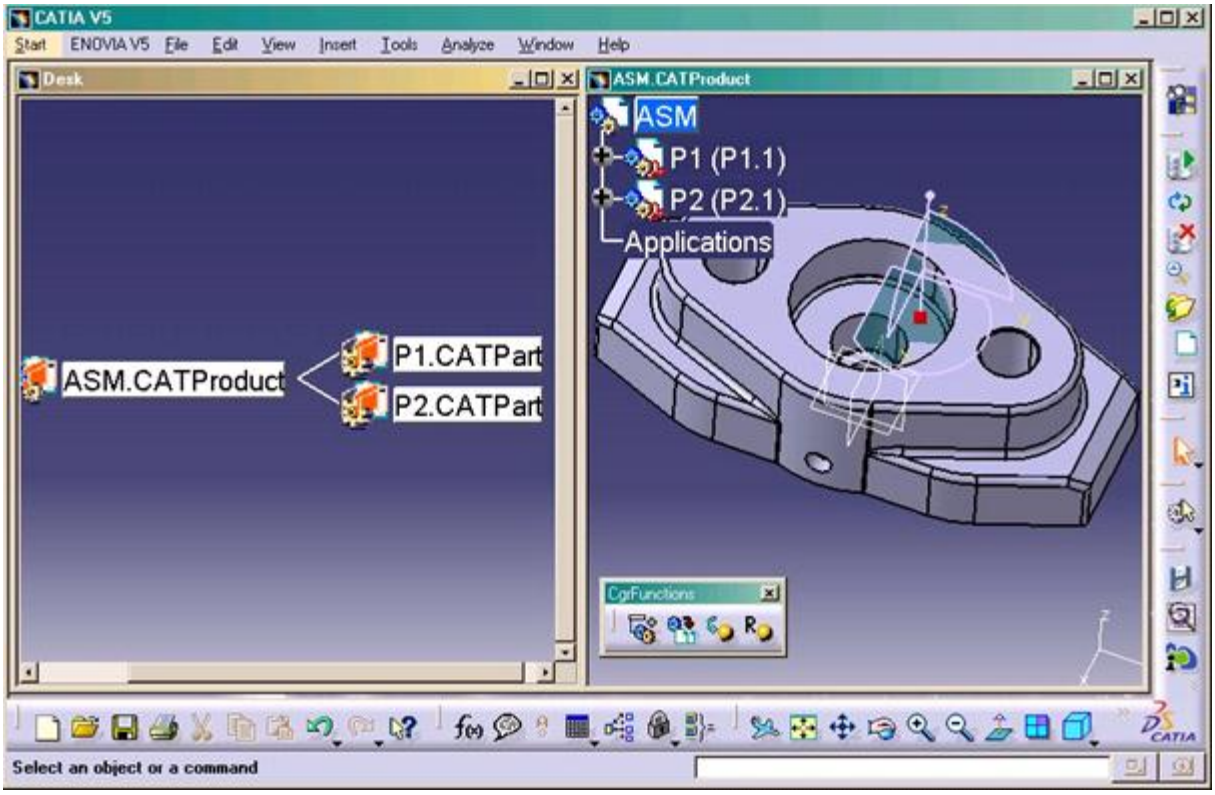


**Figure: Screenshot from the Assembly ASM**




In order to provide this functionality, the existing Save and Load commands are modified, and four new commands are provided as follows:

**Insert CGR** – Adds a CGR model to the current CATProduct.

**Save with CGR** – Saves the CGR file along with the native CATPart or CATProduct file, and establishes the relationship between them.

**Open Native File** – Opens the native CATPart or CATProduct files for the selected CGR file(s).

**Reload CGR** – Updates the selected CGR file(s) with the latest version from PLM.

## Changes to existing commands

### *Save Command*

The Save command is modified such that when a CGR file is saved, the relationship attribute between it and its parent is set to CAX-ACC-CGR, indicating that the CGR is active in the parent assembly.

If a Design already has an associated CGR file, the CGR file is created and checked in automatically.

### *Load Command*

The Load command is modified such that when any assembly is loaded, the relationship attribute between each assembly and its children is checked to determine whether the CGR or the native CATProduct or CATPart will be loaded.

### *CGR Commands*

**Table: CGR commands**

| Button | Command | Description |
|---|---|---|
|  | Save with CGR (AglSaveWithCgr) | This command creates or updates a CGR file from its corresponding native CATProduct or CATPart file, and then saves both files into a common PLM Design. This command associates the CGR and native files together, so that subsequent Save commands will know that the two are related. |
|  | Open Native File (AglOpenCgrObjects) | This command allows the user to open the native CATProduct or CATPart file, that corresponds to one or more selected CGR files. This command is initiated by first selecting the CGR file(s) in the model tree, then clicking the command. The native files are opened in separate windows from the original CGR files. The user can then, for example, replace the CGR file with the native file. |
|  | Reload CGR (AglReloadCgrObjects) | This command allows the user to update an existing CGR file with the latest version from PLM. This command is initiated by first selecting the CGR file(s) in the model tree, then clicking the command. |

# Configuration Handling – SolidWorks

## Introduction

Companies working with SolidWorks use configurations for different purposes.

One purpose is to have alternate geometric representations for the same part for use in assemblies, like an expanded or collapsed spring. From PLM point of view, there is no need to store a Design object for each representation in PLM.

The second purpose is to create parameterized designs that drive dimensions, features or structures through configurations. From a PLM point of view each of these configurations represents a separate part and therefore needs a separate Design object in PLM.

## Functionality Overview

By default the SolidWorks integration creates a Design object for each configuration in PLM if there is more than one configuration defined inside the SolidWorks file. The Default can be modified using the switch **ConfiguredDefault** in the **xPLMSolidworksConnector.xml**. In order to support the different usages of configurations, the SolidWorks connector introduces two parameters that control the representation of each configuration in PLM.

One parameter is a standard parameter, which is a general switch for all contained configurations in the SolidWorks part. By default, the parameter `Configured` is read. If this parameter is set to `No`, then no Design objects are created for any of the configurations.

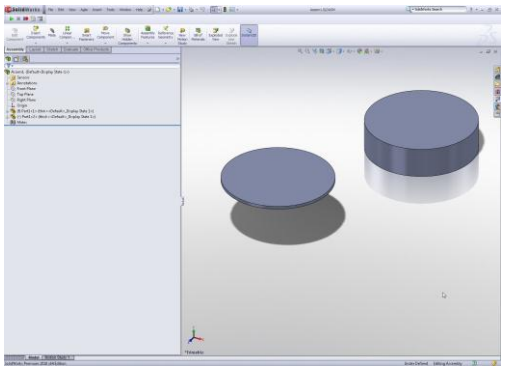
An additional configuration-specific parameter can link a specific configuration to a master configuration in the same part. Only for the master configurations a PLM object is created, all linked configurations are referenced to the master configuration. By default, the configuration-specific parameter `MasterConfig` is read. If this parameter is set and references an existing configuration in the same part, the references master configuration is used in PLM. Otherwise, a separate Design object is created for the configuration in PLM. This supports the use case of having some alternate representations and real representations mixed within one configuration table.

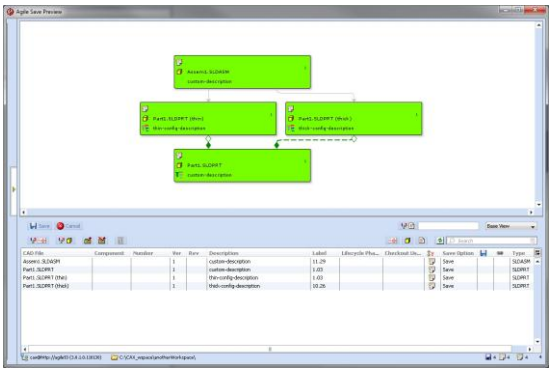
The SolidWorks property names that control the behaviour are configurable in
**XPlmSolidWorksConnector.xml**

```
<Field>
    <Name>ConfiguredProperty</Name>
    <Value>Configured</Value>
    <!--default is "Configured" -->
</Field>
<Field>
    <Name>MasterConfigProperty</Name>
    <Value>MasterConfig</Value>
    <!--default is "MasterConfig" -->
</Field>
```

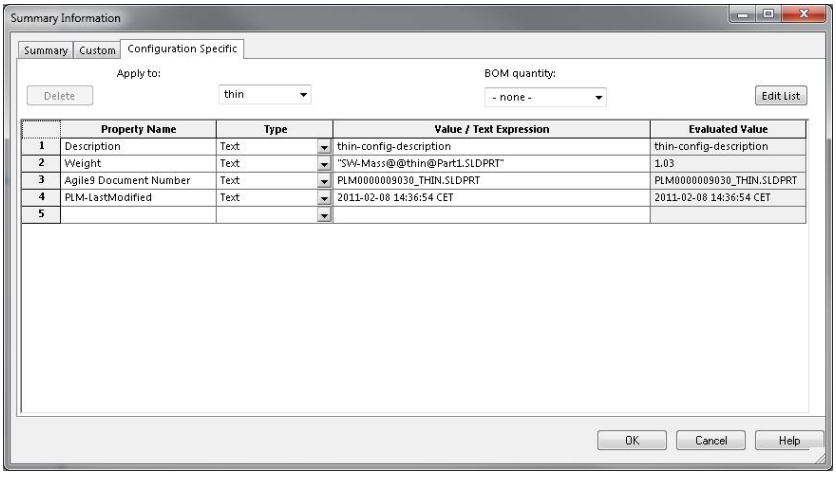
## Sample Configuration Handling

The assembly contains two configurations of the same part. Each configuration is represented with a Design object in PLM.

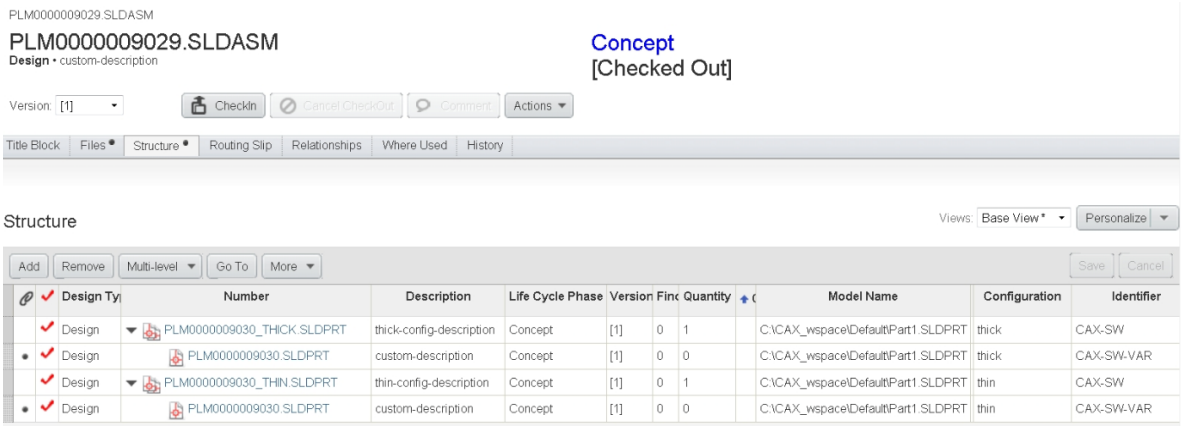




During save the configuration specific properties are filled in.

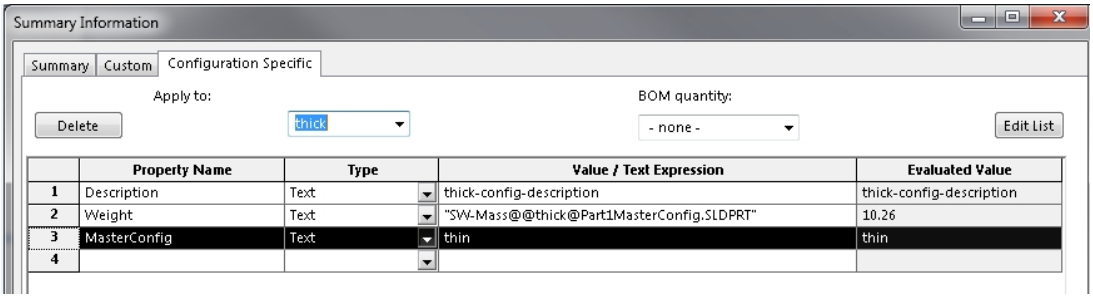



The structure in PLM looks like this.

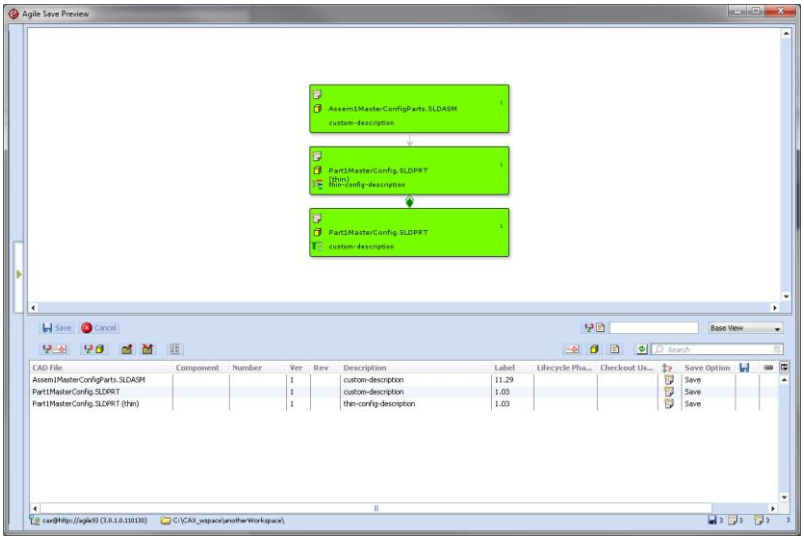

PLM0000009029.SLDASM
PLM0000009029.SLDASM
Design • custom-description
Concept
[Checked Out]
Version: [1]
CheckIn
Cancel CheckOut
Comment
Actions
Title Block
Files
Structure
Routing Slip
Relationships
Where Used
History
Structure
Views: Base View*
Personalize
Add
Remove
Multi-level
Go To
More
Save
Cancel
Design Ty
Number
Description
Life Cycle Phase
Version
Finc
Quantity
Model Name
Configuration
Identifier
Design PLM0000009030_THICK.SLDPRT thick-config-description Concept [1] 0 1 C:\CAX_wspace\Default\Part1.SLDPRT thick CAX-SW
Design PLM0000009030.SLDPRT custom-description Concept [1] 0 0 C:\CAX_wspace\Default\Part1.SLDPRT thick CAX-SW-VAR
Design PLM0000009030_THIN.SLDPRT thin-config-description Concept [1] 0 1 C:\CAX_wspace\Default\Part1.SLDPRT thin CAX-SW
Design PLM0000009030.SLDPRT custom-description Concept [1] 0 0 C:\CAX_wspace\Default\Part1.SLDPRT thin CAX-SW-VAR

## Sample Master Configuration

The assembly contains two configurations of the same part. One configuration is linked to the other configuration using the configuration-specific property MasterConfig. The example below links the configuration *thick* to *thin*.




During the save process the two configurations are treated as one object. Only the master configuration (*thin*) is created in PLM. The save preview is shows only the master configurations.




The structure in PLM looks like this and sums up the quantity of *thick* and *thin*.