Connector for Creo Parametric

Install and Administration Guide

3DEXPERIENCE R2022x

DASSAULT SYSTEMES | The 3DEXPERIENCE Company

Prepared by International TechneGroup Inc.

Table of Contents

Overview

3DEXPERIENCE - MCAD integrations based on X-CAD Design management offers a flexible and powerful way to manage and control information between 3DEXPERIENCE COLLABORATION PLATFORM and various CAD systems. They allow companies to establish a real-time seamless link across Engineering and Design organizations. The resulting up-to-date information drives product delivery for faster time-to-market.

This Installation and Administration Manual provides an overview of the installation and administration tasks that must be performed for the 3DEXPERIENCE Creo Parametric integration that works with the IEFClient and the X-CAD Design applications. This manual is the starting point for the ITI - developed Creo integration.

An 3DEXPERIENCE Creo Parametric integration installation and configuration is performed in four sequential steps. This document is broken into four main sections, one for each step:

- Install the Server side Schema elements required for Creo Parametric integration.
  The installer name is: *ConnectorforCreoParametricServer-V6R2022x.AllOS.tar.gz*

- Install the Creo Parametric License Key Generator to generate the license key required for installing the Connector for Creo Parametric client
  The installer name is: *ConnectorforCreoParametricLicenseKeyGenerator-V6R2022x.AllWindows.zip*

- Install the Connector for Creo Parametric Client.
  The installer name is: *ConnectorforCreoParametricClient-V6R2022x.Windows64.zip*

- 3DEXPERIENCE Creo Parametric Integration Administration and Configuration
  This section provides information about the Administration and Configuration of the Server side schema and UserExits functionality.

Prerequisites

Server side prerequisites

- 3DEXPERIENCE Server Setup
- X-CAD Design 3DEXPERIENCE R2022x GA
- Server-Side Integration Schema Installation (GA)
- Configuration of users with required roles and licenses – Requires licenses and roles
- Configuration of required attributes and types
- Licenses required for performing integration activities:
    - DEP - Connector for Creo Parametric
    - CDR - Component Designer
    - CSV - Industry Innovation
    - IFW - IFWE Compass

Client side prerequisites

- Platforms

  Connector for Creo Parametric is supported on the following platforms.
    - Windows 10 Professional and Enterprise 64 Bit

- Creo Parametric

  Following are the Creo Parametric versions supported.
    - Creo 5.0 (64 Bit)
    - Cre0 6.0 (64 Bit)
    - Creo 7.0 (64 Bit)

- IEFClient

  The Connector for Creo Parametric client requires the 3DEXPERIENCE Integration Exchange Framework Client (IEF) - 3DEXPERIENCE R2022x installed on the client machine.

  IEFClient has additional requirements such as JRE version. Please refer to X-CAD Design documentation for details.

Note: Requires PTC Pro/TOOLKIT product (Application Programming Toolkit) or Creo Elements/Pro TOOLKIT Customization API, Product ID PTN-3119-L

Installation of 3DEXPERIENCE Connector for Creo Parametric Server Schema

Schema Installation

The installer of Connector for Creo Parametric Server, ConnectorforCreoParametricServer-V6R2022x.AllOS.tar.gz installs the Schema elements required for the Creo Parametric Connector to be functional.

Extract the contents of the tar file, ConnectorforCreoParametricServer-V6R2022x.AllOS.tar.gz on the 3DEXPERIENCE server machine.

A tcl script must be run from the mql command prompt to install schema for Connector for Creo Parametric. This script, ProEIntegInstallSchemaMain.tcl, creates/updates default schema and a mapping object consistent with this default schema.

The default schema consists of a set of pre-defined attributes, types, relationships, and policies the integration can use “as-is”. The script ProEIntegInstallSchemaMain.tcl creates the default schema and a mapping object.

- Run ProEIntegInstallSchemaMain.tcl with the following steps:

  Launch mql command prompt from the 3DSpace server folder location

  cd <Contents of Schema > [change directory to the main tcl]

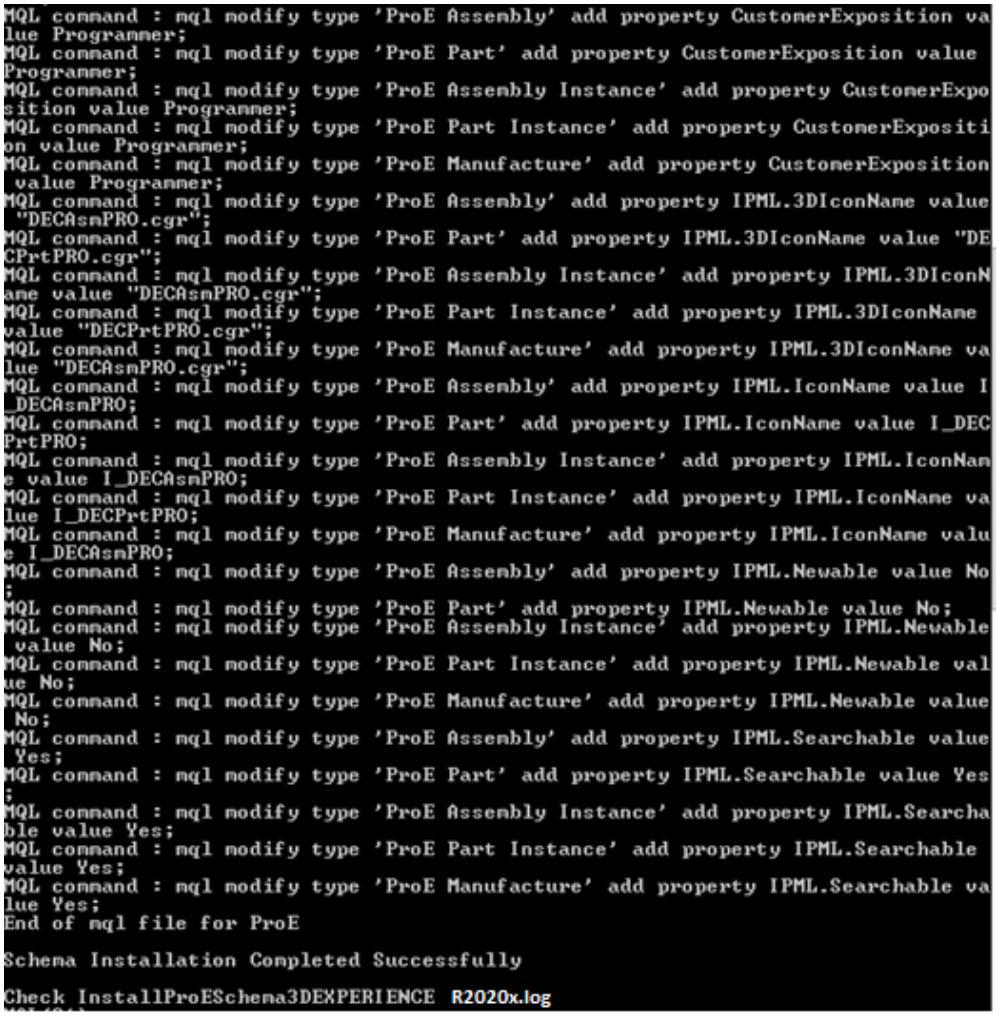
  mql [start mql from server (..\server\bin\winb_64\code\bin) folder]

  set context user creator vault ‘eService Production’ [set context before install]

  run ProEIntegInstallSchemaMain.tcl [run the tcl script]

The log file, InstallProESchemaV6R2022x.log, contains errors and warnings during the schema install.

Following screenshot provides the output of successful execution of the Server schema installer.



Picture: Output of the Connector for Creo Parametric Server schema installation

The 'ProENGINEER User' Role

Connector for Creo Parametric creates a new role called 'ProENGINEER User' as part of Schema installation**. All users who run this Connector for Creo Parametric must have the role of 'ProENGINEER User' added to their assignments.** The administrator must add this role to every user of the Connector for Creo Parametric otherwise the user will not be able to use the Connector for Creo Parametric.

Users must be assigned to use the Connector for Creo Parametric. The administrator, who must have the role of 'Integration Administrator', must assign users to the Connector for Creo Parametric. Please refer to the ***X-CAD Design User Guide*** for more information.

Connector for Creo Parametric License

Connector for Creo Parametric uses Connector for Creo Parametric license. **All users who run Connector for Creo Parametric must have Connector for Creo Parametric license.** The administrator must assign this license to every user of the Connector for Creo Parametric; otherwise, the user will not be able to use the Connector for Creo Parametric. Please refer to the ***X-CAD Design User Guide*** for more information.

Creating Collaborative Space and assigning user

Admin must perform the following additional steps on the Server to make the integration work with default OOTB configuration.

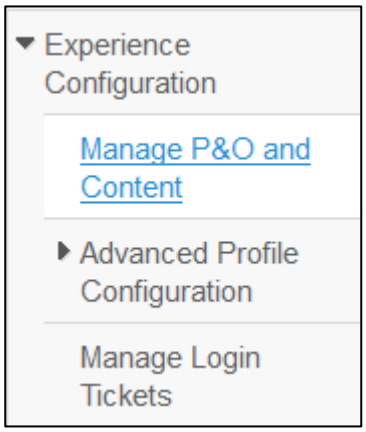
Step 1: Assign license to PLMADM

Login to MQL as creator and then execute following statements.

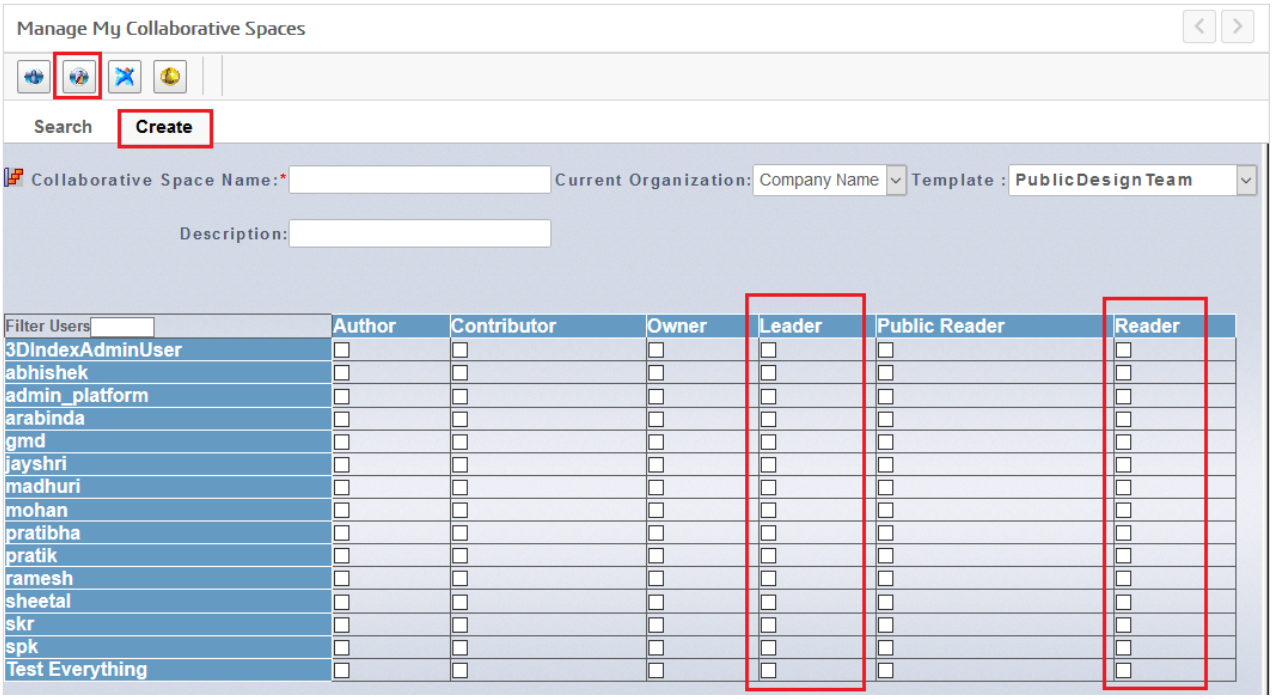
```
modify product CSV add person "PLMADM";
modify product CDR add person "PLMADM";
modify product DEP add person "PLMADM";
modify product IFW add person "PLMADM";
```

Step 2: Create Collaborative space and assign users with roles like **Leader**, **Reader**, etc.

- Go to the web browser and login to 3DEXPERIENCE as PLMADM
- From the 'Experience Configuration' section, select "Manage P&O and Content"



- Click on the second icon (As highlighted below) and then create a collaborate space.



- Add users with Author, Leader and Reader role as shown above. Only people with Leader role in "Collaborative Space" can create PROE type objects via integration

- Click on 'Create' button to create the collaborative space.

At this point, the Connector for Creo Parametric needs to be tested to make sure it is functional with the OOTB configurations. After proving out the integration against the default schema is functional, modifications can be made to meet individual business practices and processes and to fit into an overall schema.

3DEXPERIENCE Connector for Creo Parametric License Key Generation

Overview

The installer ‘ConnectorforCreoParametricLicenseKeyGenerator-V6R2022x.AllWindows.zip’ is used for generating the License Key which is required during the install process of the Connector for Creo Parametric client.

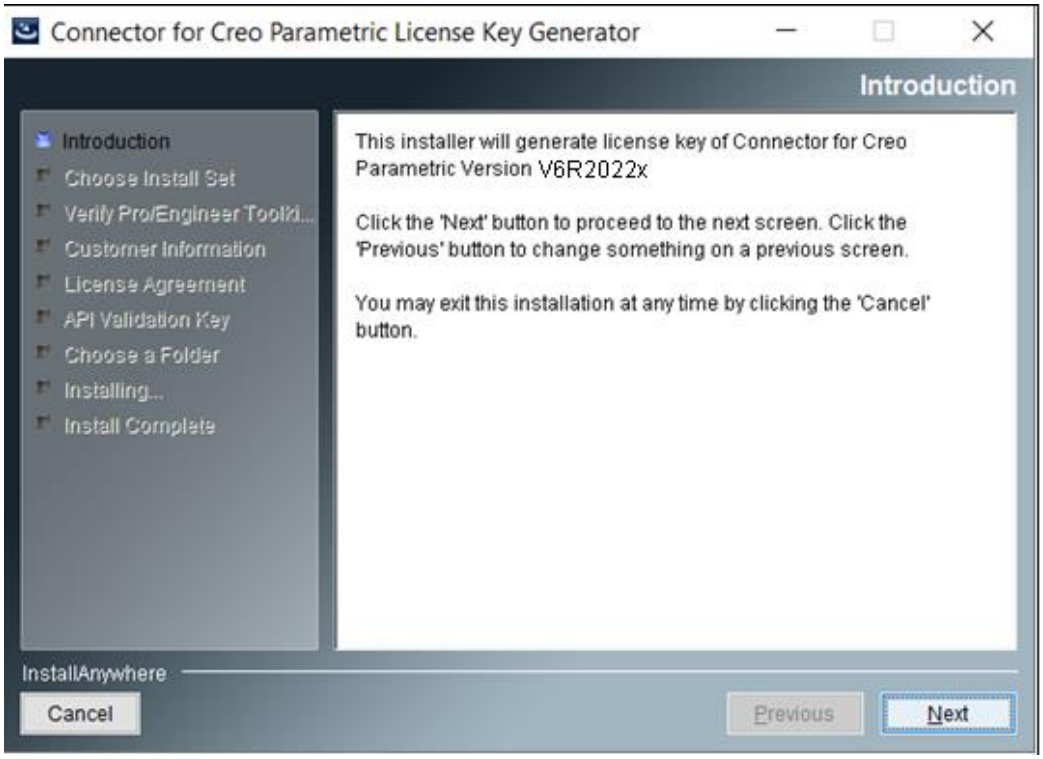
For generating the license key for Connector for Creo Parametric, the machine on which the installation is performed should have access to the Creo License Server containing Pro/TOOLKIT license.

The administrator needs to perform the following install steps to generate the License Key.

Installation of Creo Parametric License Key generator

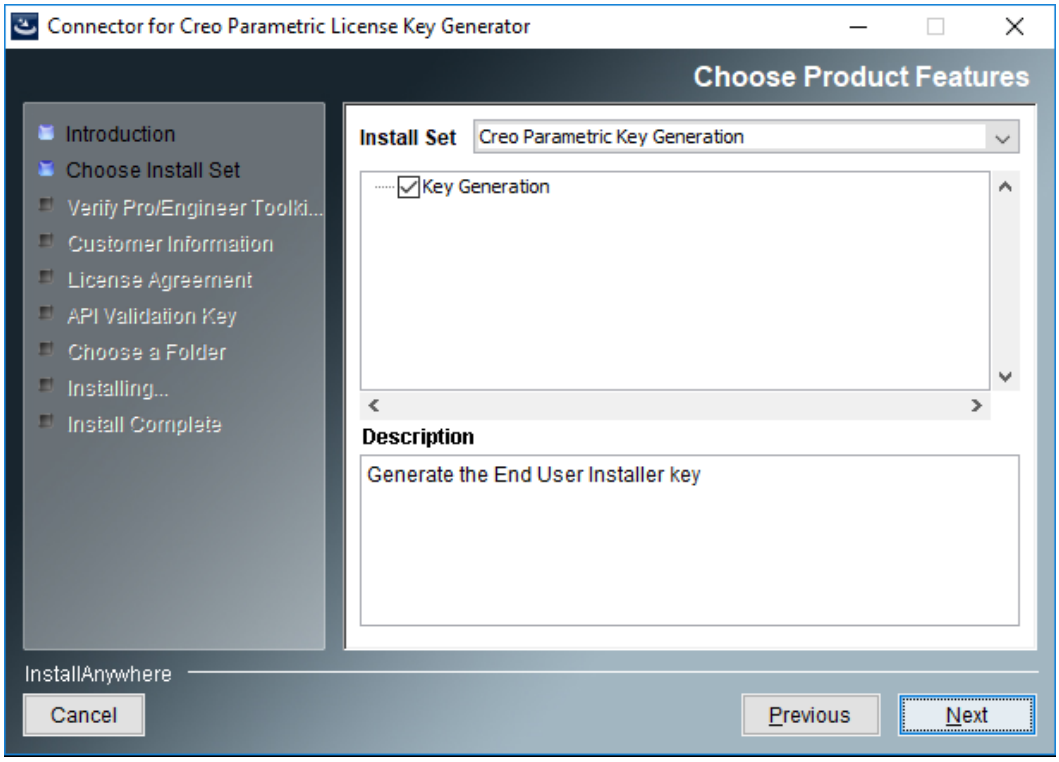
Extract installer from ConnectorforCreoParametricLicenseKeyGenerator-V6R2022x.AllWindows.zip and run the license key installer on the windows OS machine.

The installer will start with the Introduction window displayed with information regarding the install process and version.



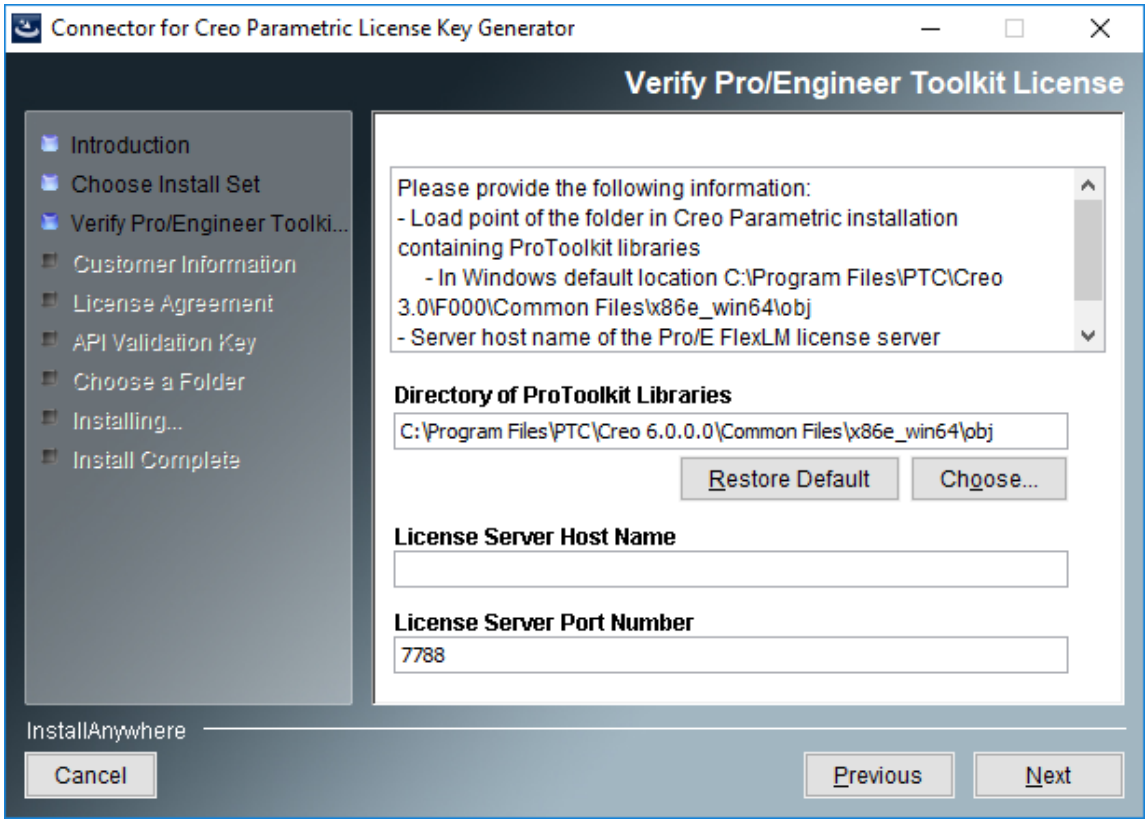
Select **Next.**

The choose product features window basically shows that Key Generation process is selected.

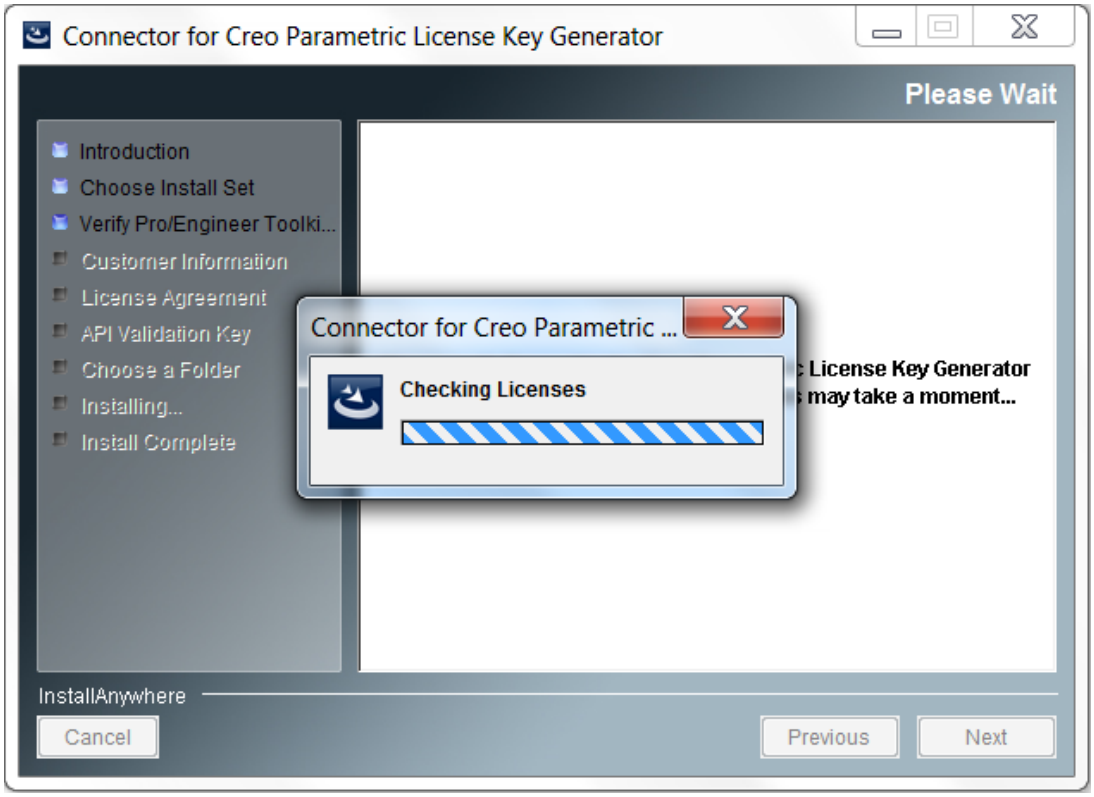


Select **Next.**

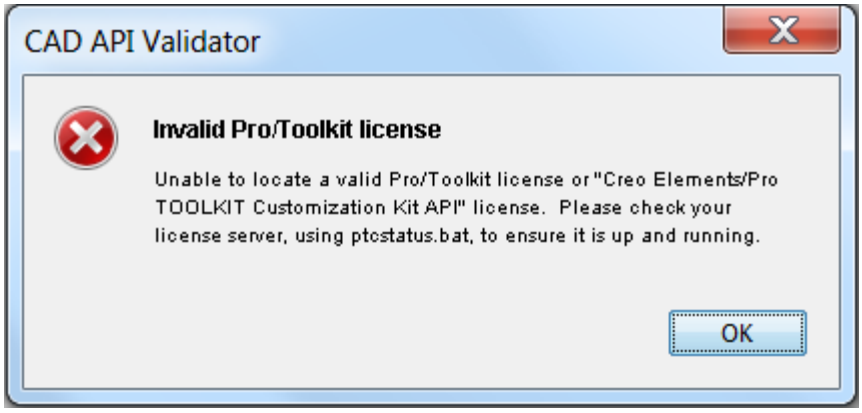
The installer will verify Pro/Toolkit license. This verification is done via FlexLM. The installer is looking for the directory installed by Creo Parametric that contains the lmutil program, license server hostname and port.



Enter the Creo Parametric lmutil directory, license server hostname and port and select **Next.** The installer will verify the licenses.

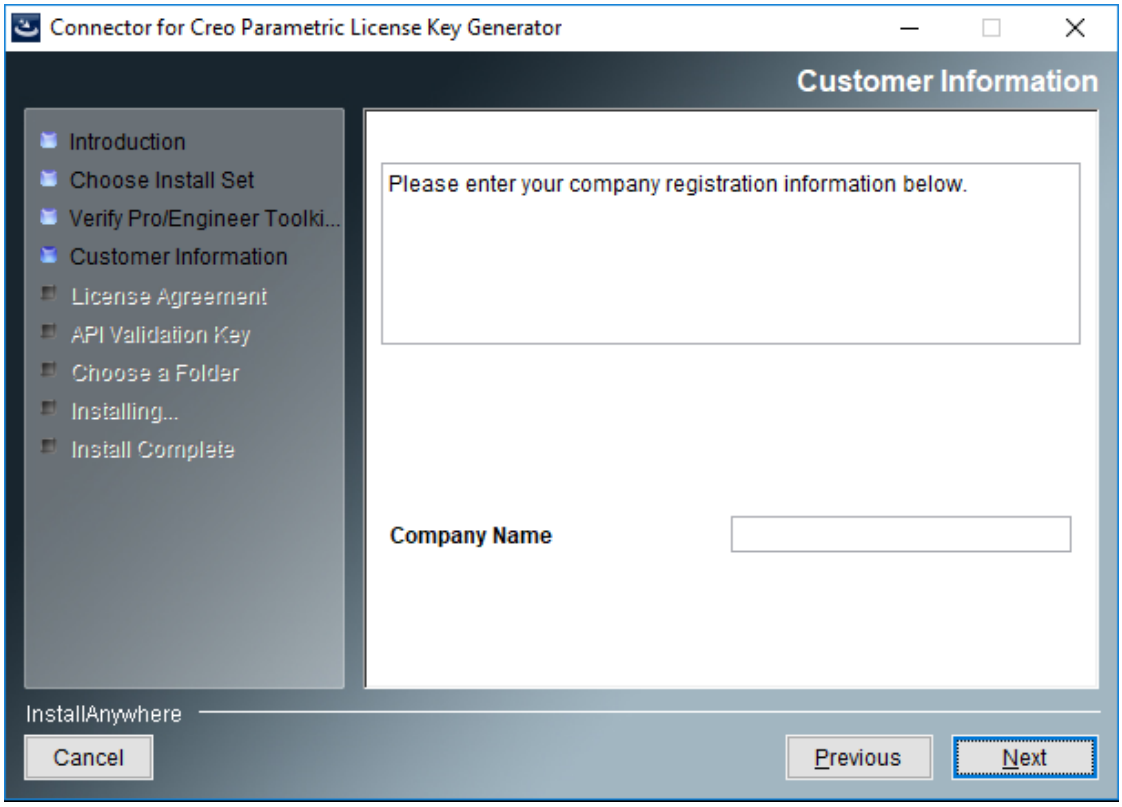


If no Pro/Toolkit license is available, the following panel is displayed:



Once **OK** is selected, the installer returns to verify Pro/Toolkit License screen. The installer cannot find the Pro/Toolkit license. Please verify that it is available and can be accessed by this computer. If problems persist, contact your Creo Parametric administrator. Your company must have a valid Pro/Toolkit license to generate license key of Connector for Creo Parametric.

Once the license validity is checked, the installer will prompt for the company name:

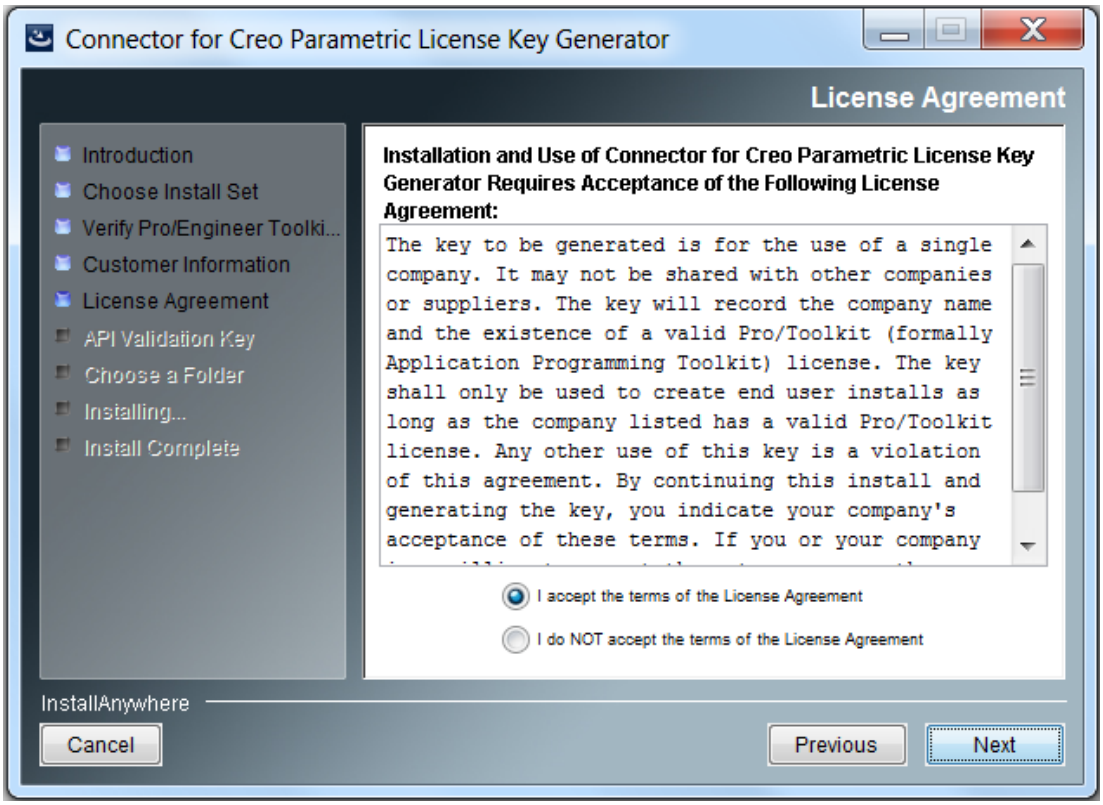


Enter the company name (owner of the Pro/Toolkit license) and select **Next.**

The next panel will display a license agreement. This is the terms and conditions on the key that will be generated by the installer. The license text states:

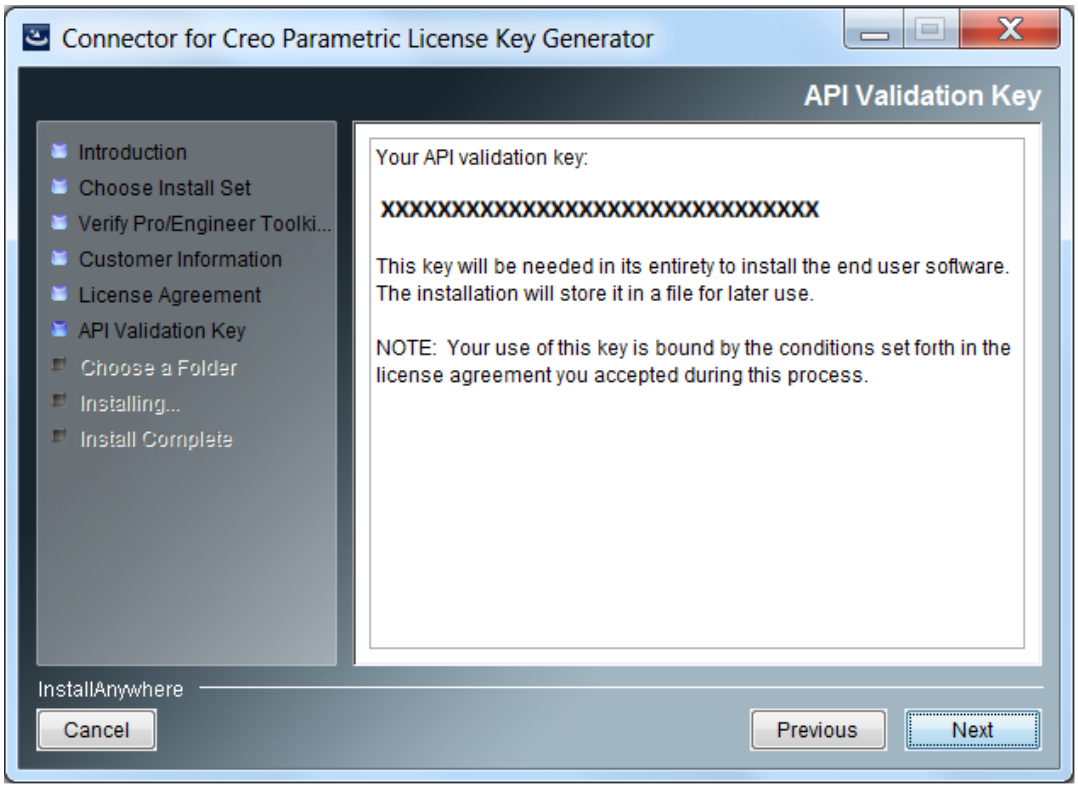
“The key to be generated is for the use of a single company. It may not be shared with other companies or suppliers. The key will record the company name and the existence of a valid Pro/Toolkit (formally Application Programming Toolkit) license. The key shall only be used to create end user installs as long as the company listed has a valid Pro/Toolkit license. Any other use of this key is a violation of this agreement. By continuing this install and generating the key, you indicate your company's acceptance of these terms. If you or your company is unwilling to accept these terms, press the cancel button to cancel the install.”

In order for a company to legally possess the Connector for Creo Parametric, they must possess a valid Pro/Toolkit license.



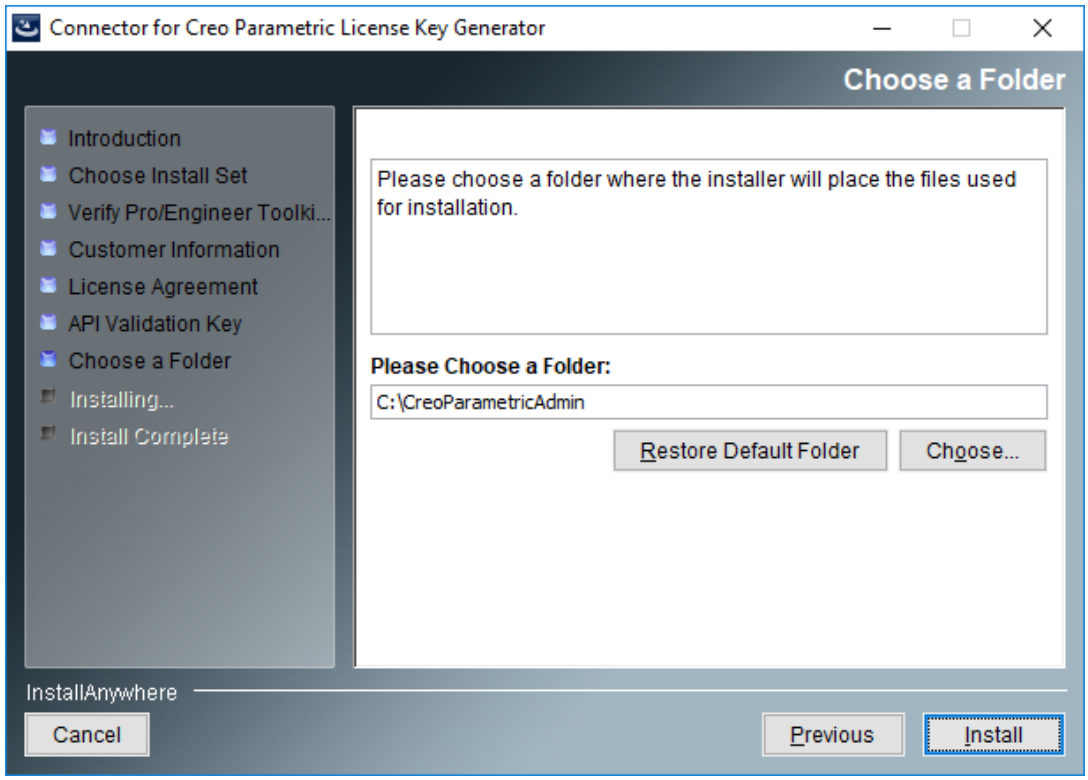
To continue the install, accept the agreement and select **Next**.

The installer will generate a key needed for all end user installs. The key can be copied from this window and emailed to end users. It will also be saved in a file in the validation directory. The Administrator might want to note down the license key, as this key is needed later to install client installers. If at any time there is a question about license validity or the key gets lost, this installer can be run to this point to check and generate a new key. Canceling at this point will NOT produce the text file.

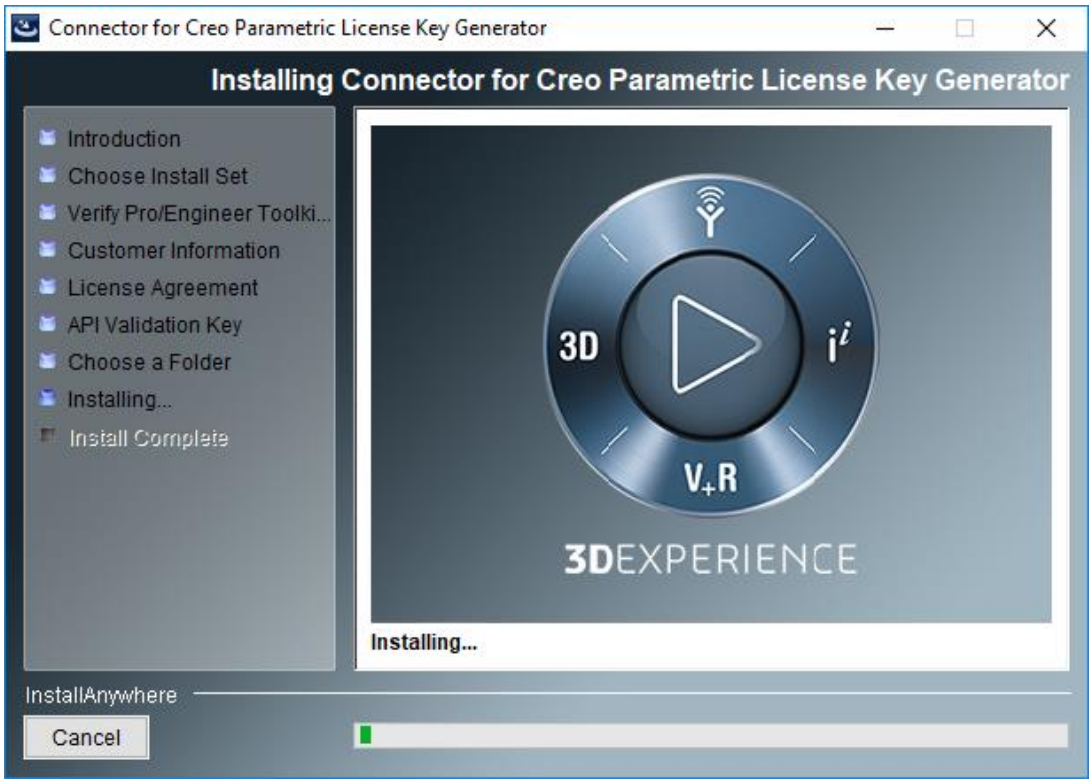


Select **Next** to proceed to identify the install folder.

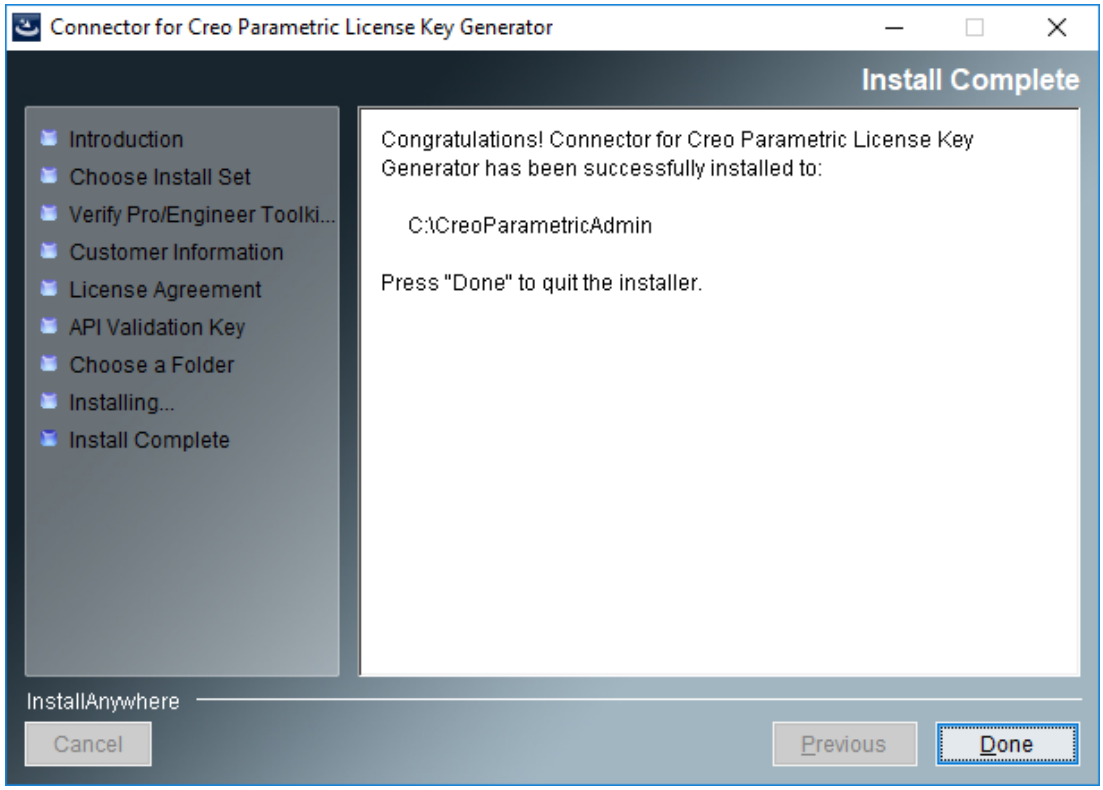
On the following panel, choose a folder for the installer output.



Select **Install** to complete the install process. On selecting Install following panel is displayed.



On successful completion, the following panel will be displayed:



Select **Done** to complete the install.

Install directory contents

The install directory will contain a subset of the following files/directories:

- installer.properties- a starting point file for generating input for a SILENT installer.
- installerinput.log- a log of user inputs and messages from the install process.
- MXPRO_LICENSE.txt- license key.

3DEXPERIENCE Connector for Creo Parametric Client

Overview

This section talks about the steps involved in installing the 3DEXPERIENCE R2022x Connector for Creo Parametric Client.

The installer to be run is: *ConnectorforCreoParametricClient-V6R2022x.Windows64.zip*.

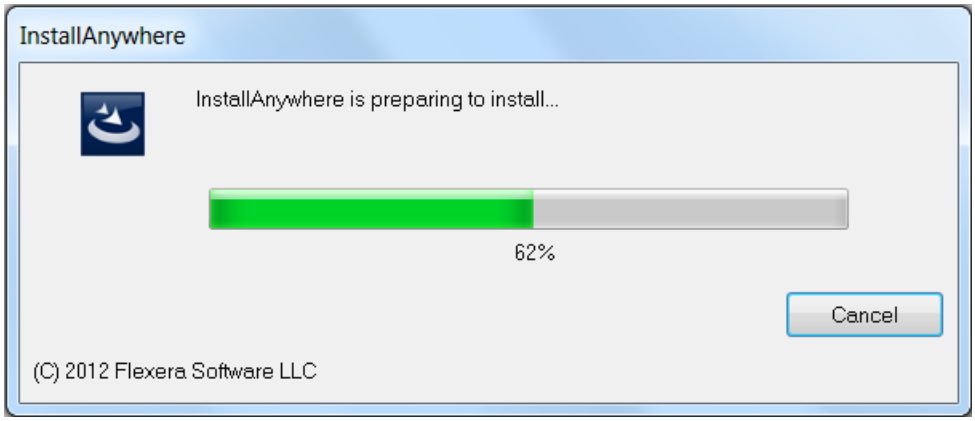
The Connector for Creo Parametric license generated by using the Connector for Creo Parametric License Key Generator is required to continue with the installation.

Installation of Connector for Creo Parametric Client

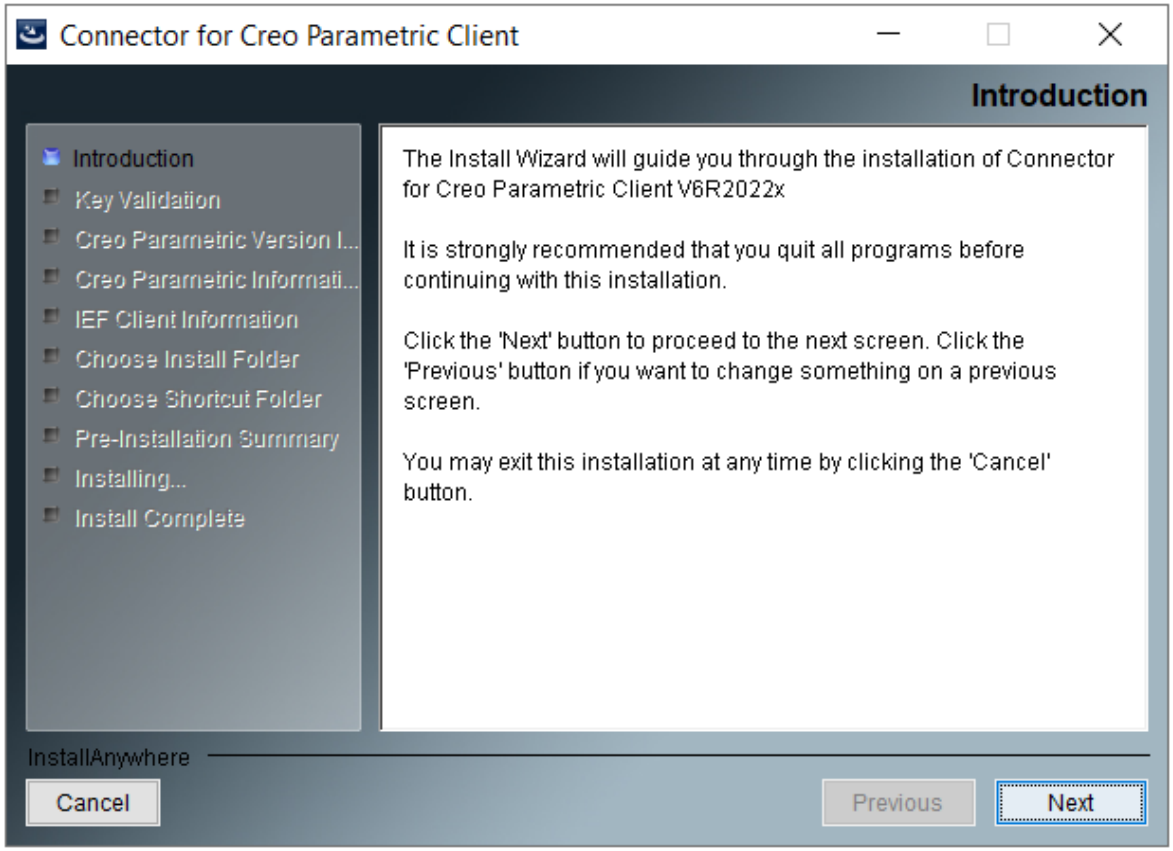
Extract the installer from zip or tar file, ConnectorforCreoParametricClient-V6R2022x.Windows64.zip.

Double Click the installer executable, ConnectorforCreoParametricClient-V6R2022x.Windows64.exe

The install process will start.

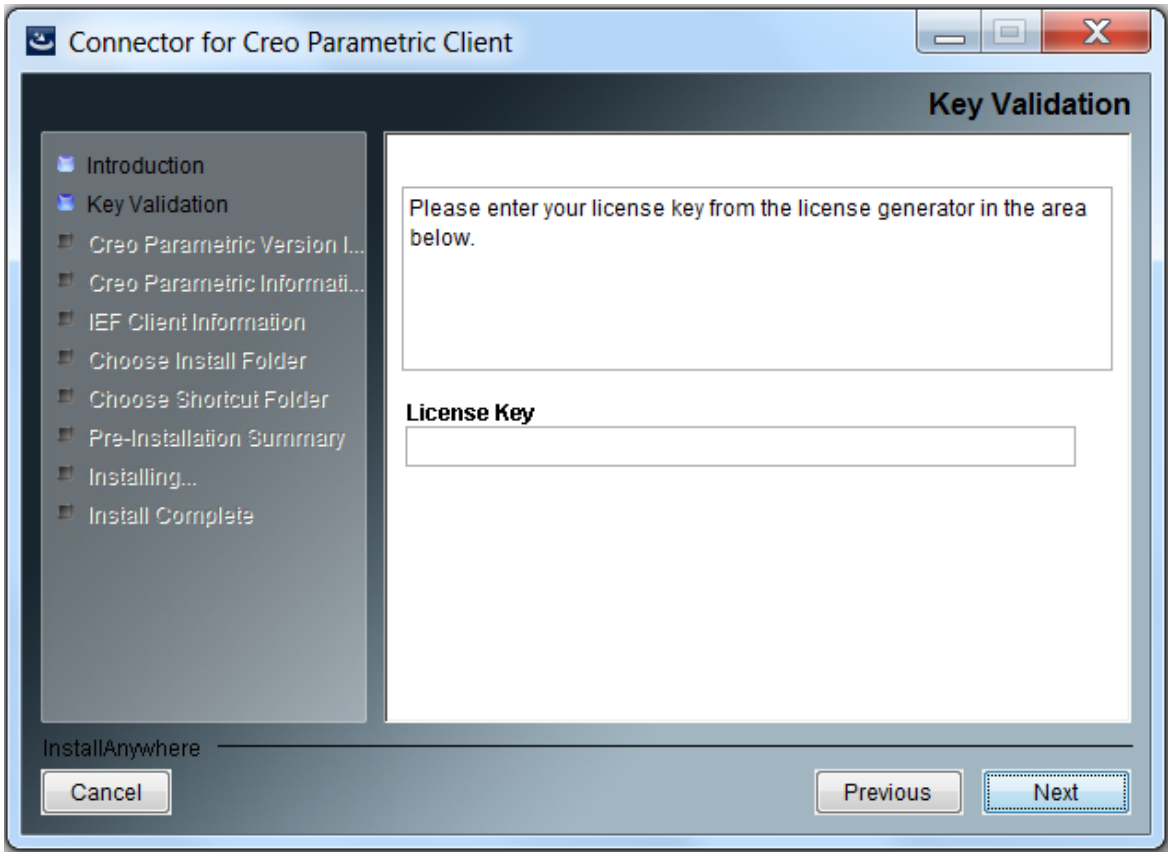


The Introduction window displays information regarding the install process



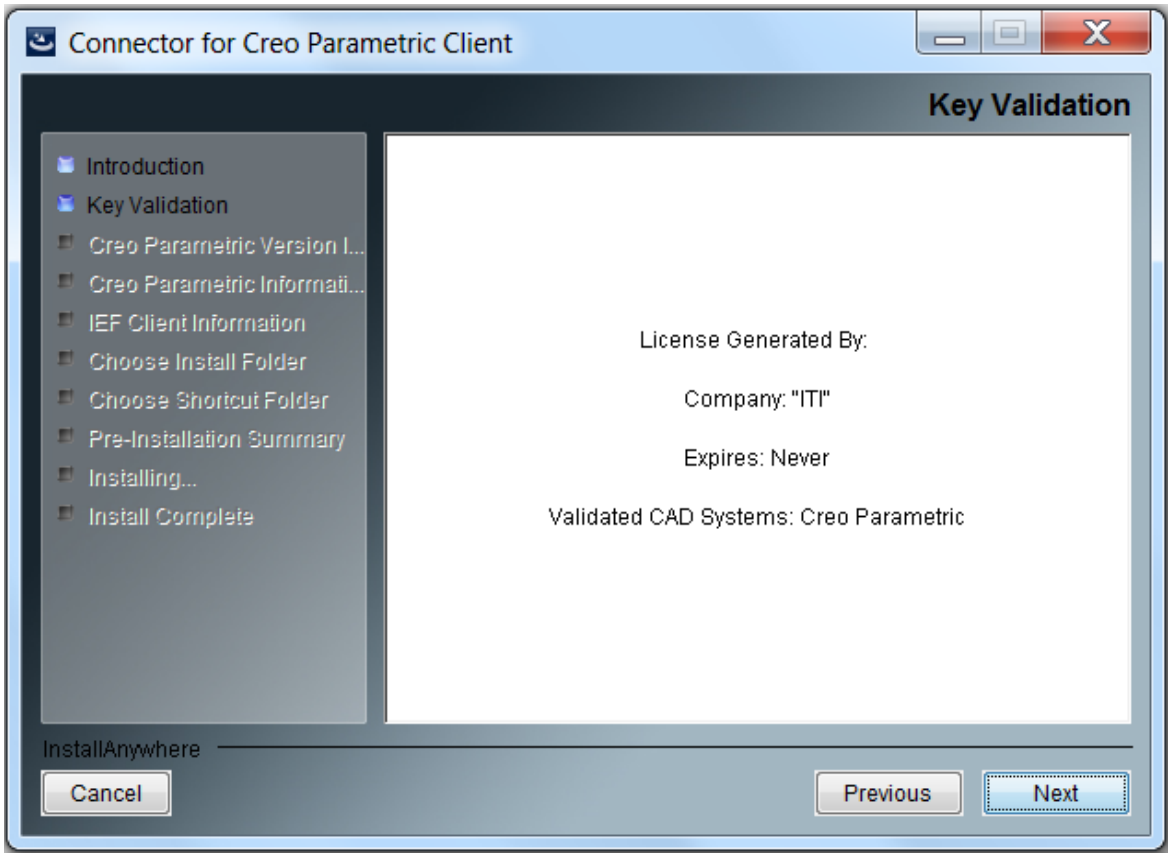
Select **Next**

You are prompted to enter License Key. This key is generated when the admin installer is run. Obtain this key from your administrator who installed the server portion of the Creo Parametric Connector.



Enter License Key
Select **Next**

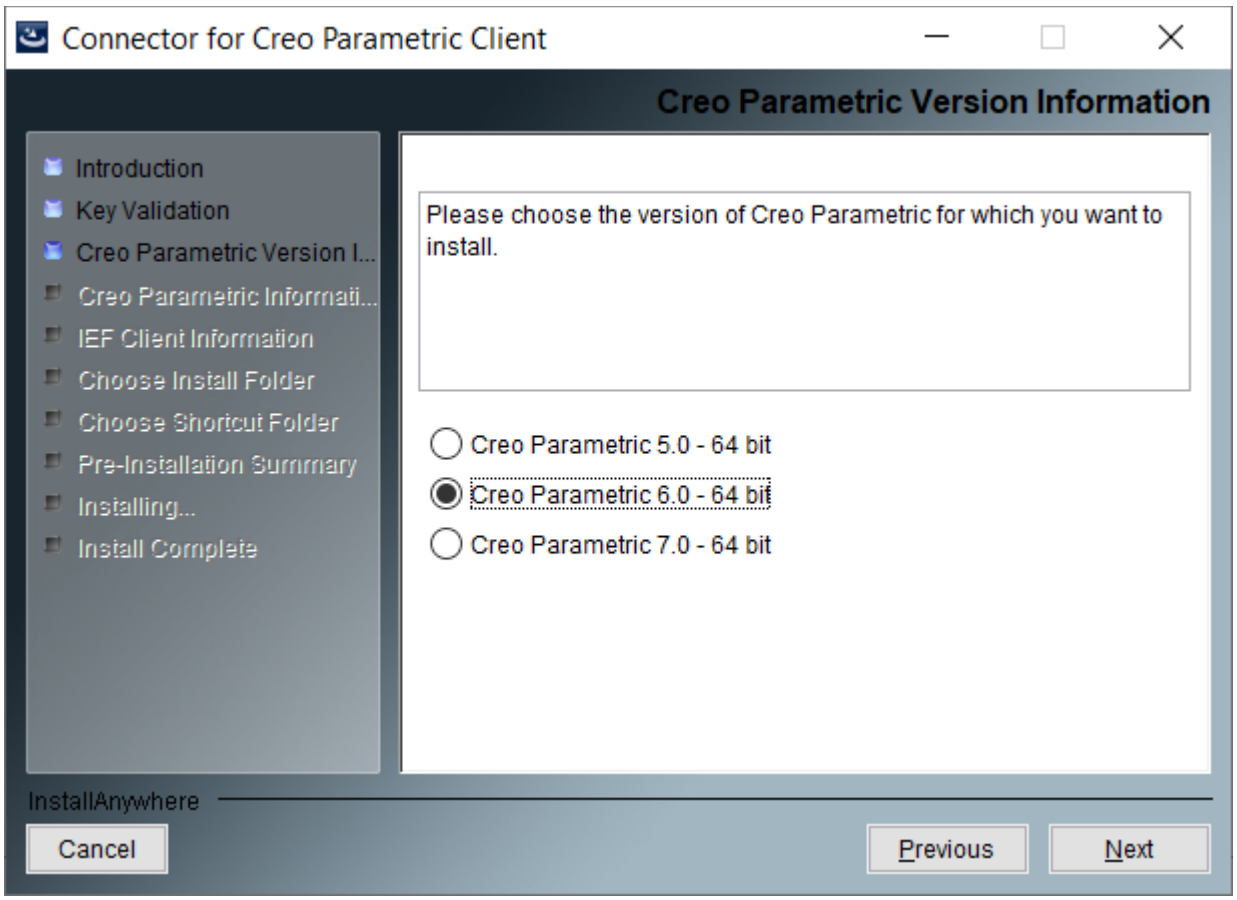
This panel shows that license was successfully applied.



Select **Next**

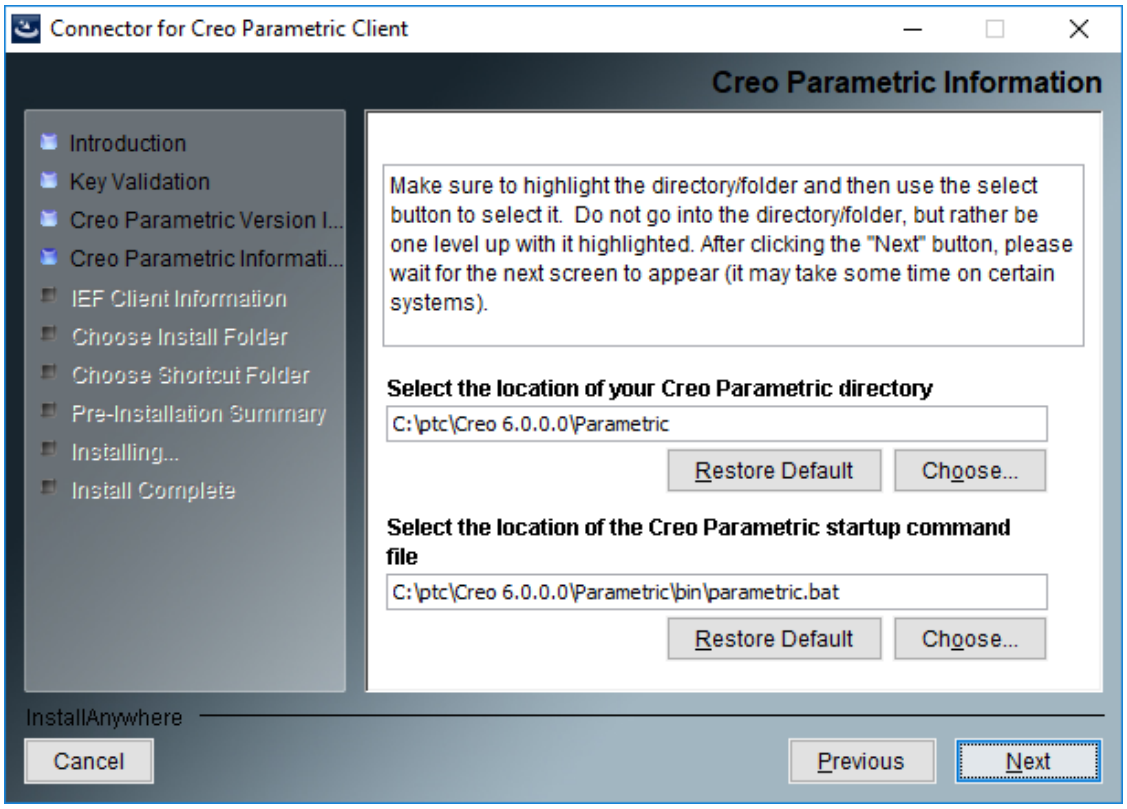
If the key is not valid, an error message is displayed (as shown here). When this occurs, select OK, re-verify and reenter the key. If the error persists, consult with the administrator. The end user code cannot be installed without a valid key.

Once through the key validation, the installer prompts for the version of Creo Parametric. Make sure to select the version you are using. If you make the wrong selection and select next the installer must be restarted in order to make a different selection.



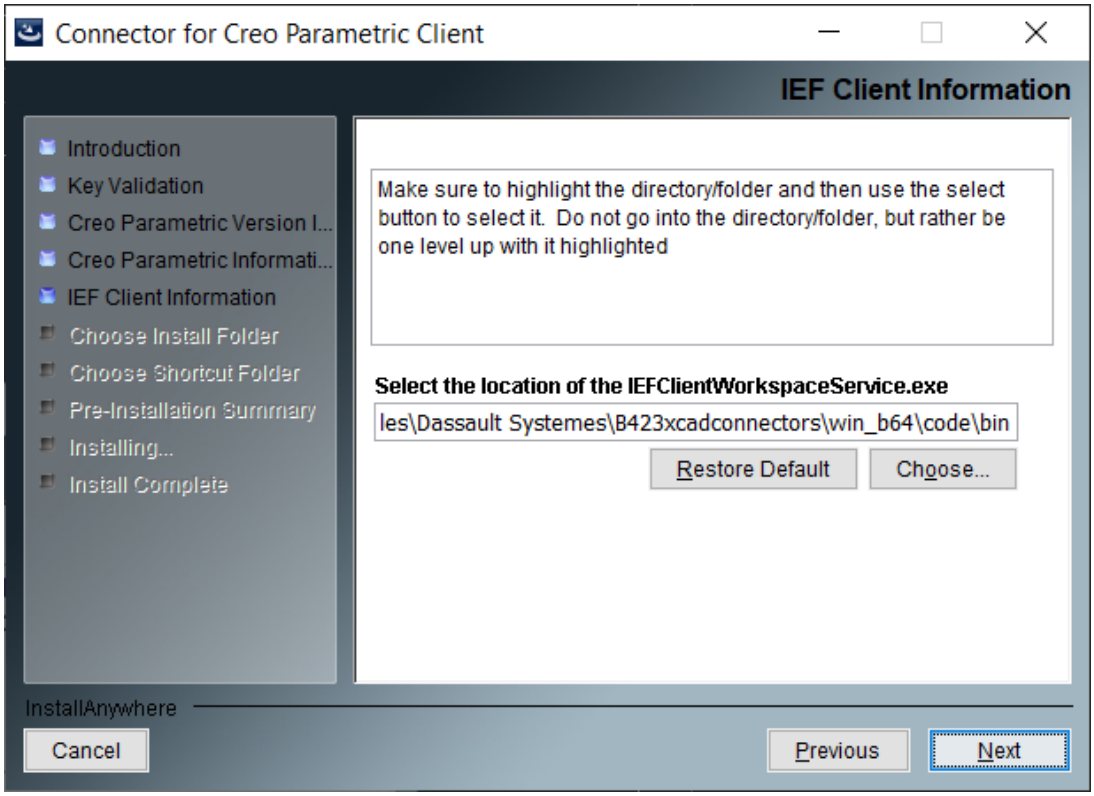
Select a version to install, then select **Next.**

Creo Parametric determines at runtime, what mode to run in, therefore the user needs to select the compatible mode of the Connector for Creo Parametric to install.
Once you have chosen the version of Creo Parametric for which you want to install, the installer prompts for the location of Creo Parametric and the Startup Command.



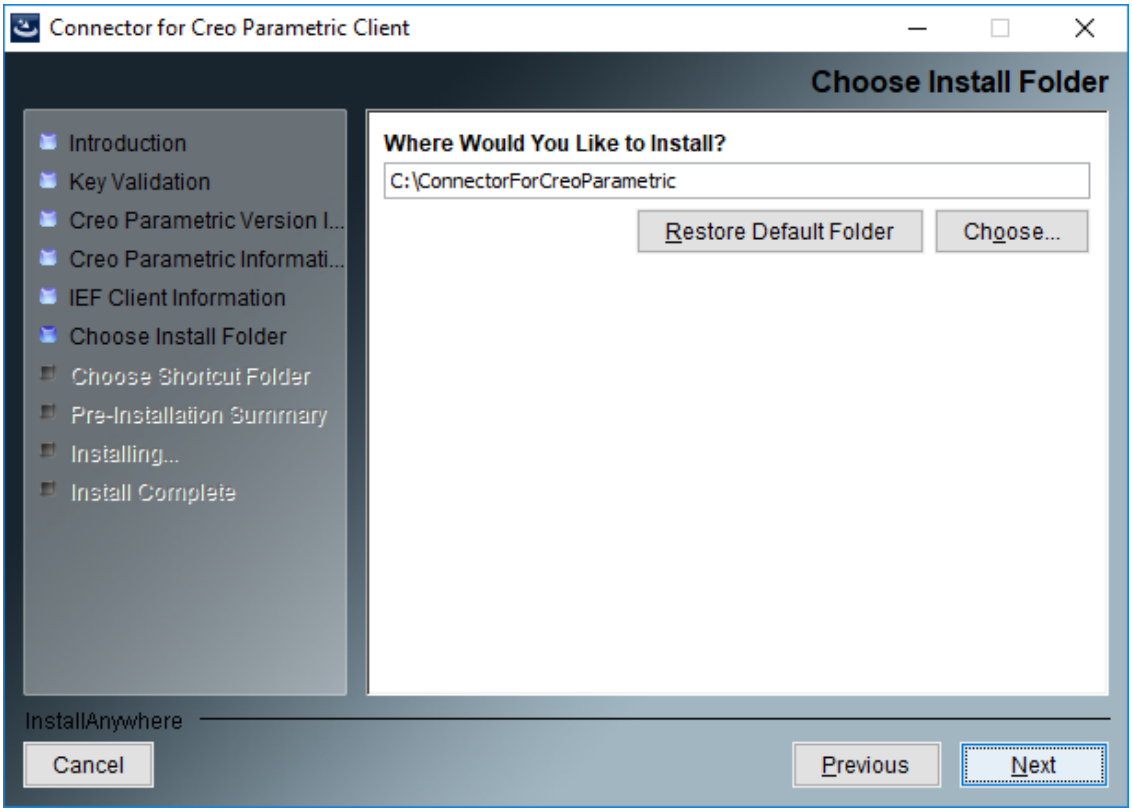
Enter the location of Creo Parametric
Select your Creo Parametric Startup Command file (The location of this file is dependent upon what was specified when installing Creo Parametric)
Select **Next**.

The installer prompts to select the location for IEF Client directory.



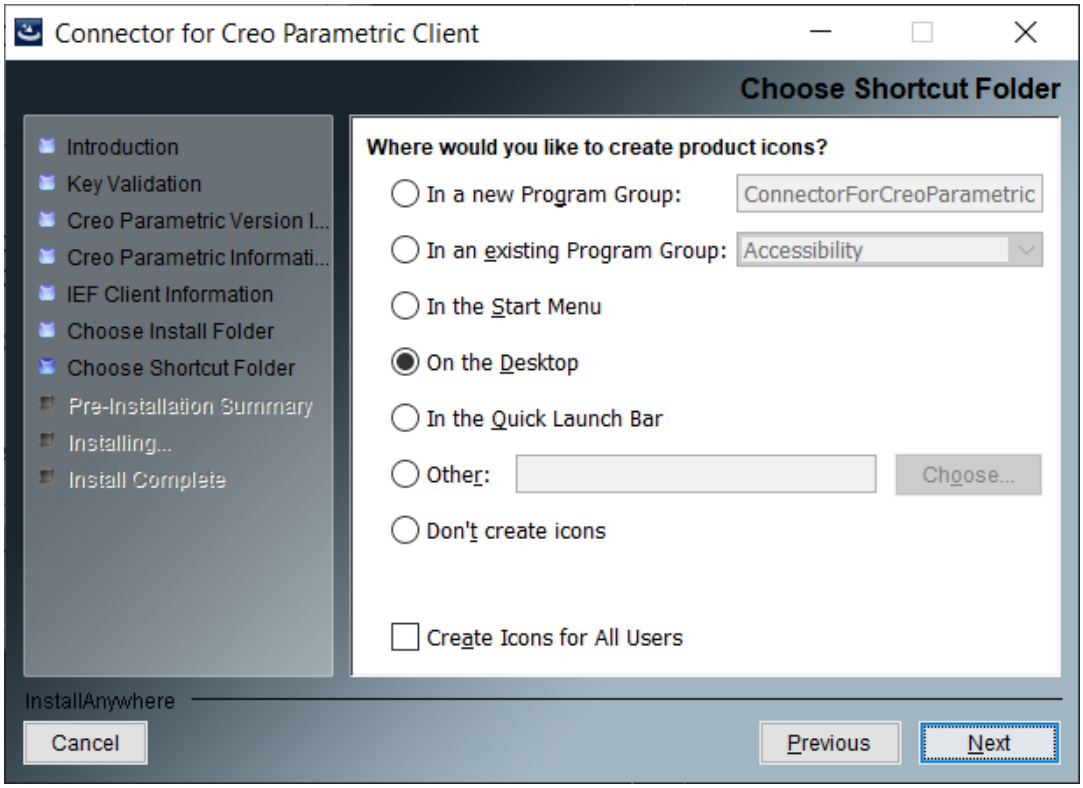
Enter the location of IEF Client directory. This is the directory that contains IEFClientWorkspaceService.exe.
Select **Next**.

The Install Location window prompts for the location where the actual files will be installed locally.

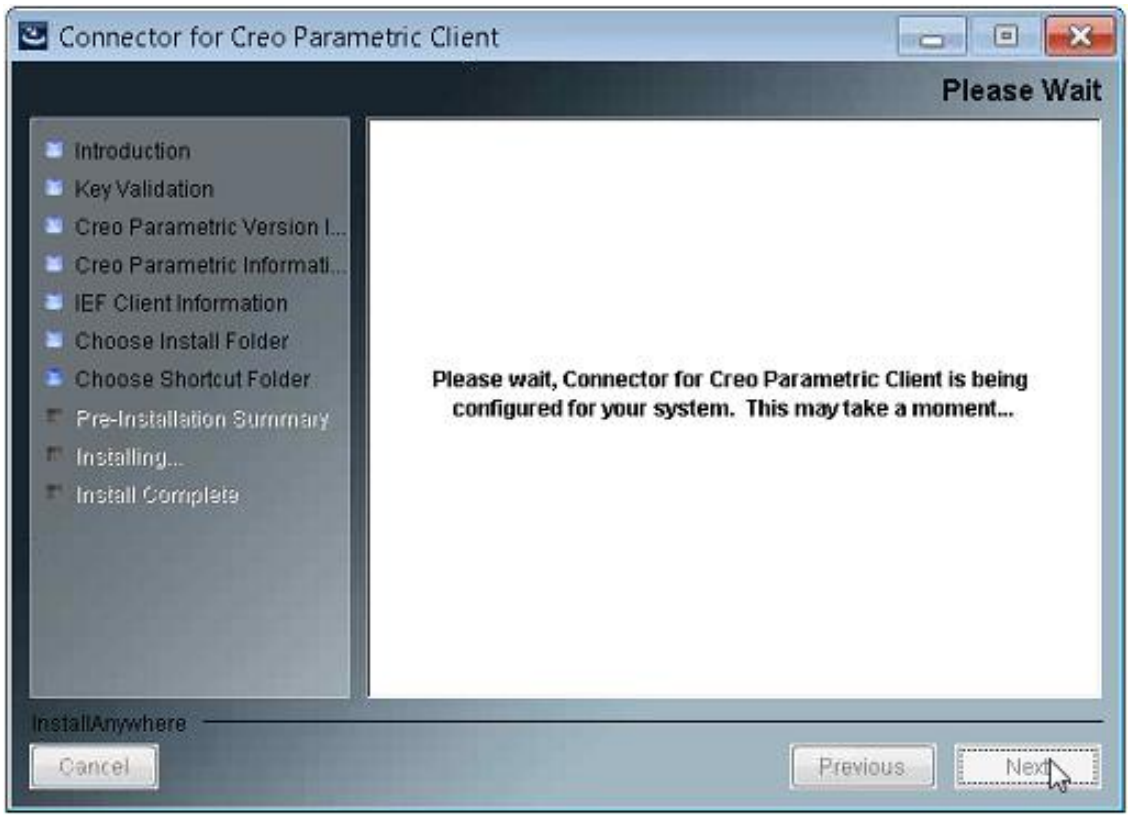


Enter the location to install
Select **Next**

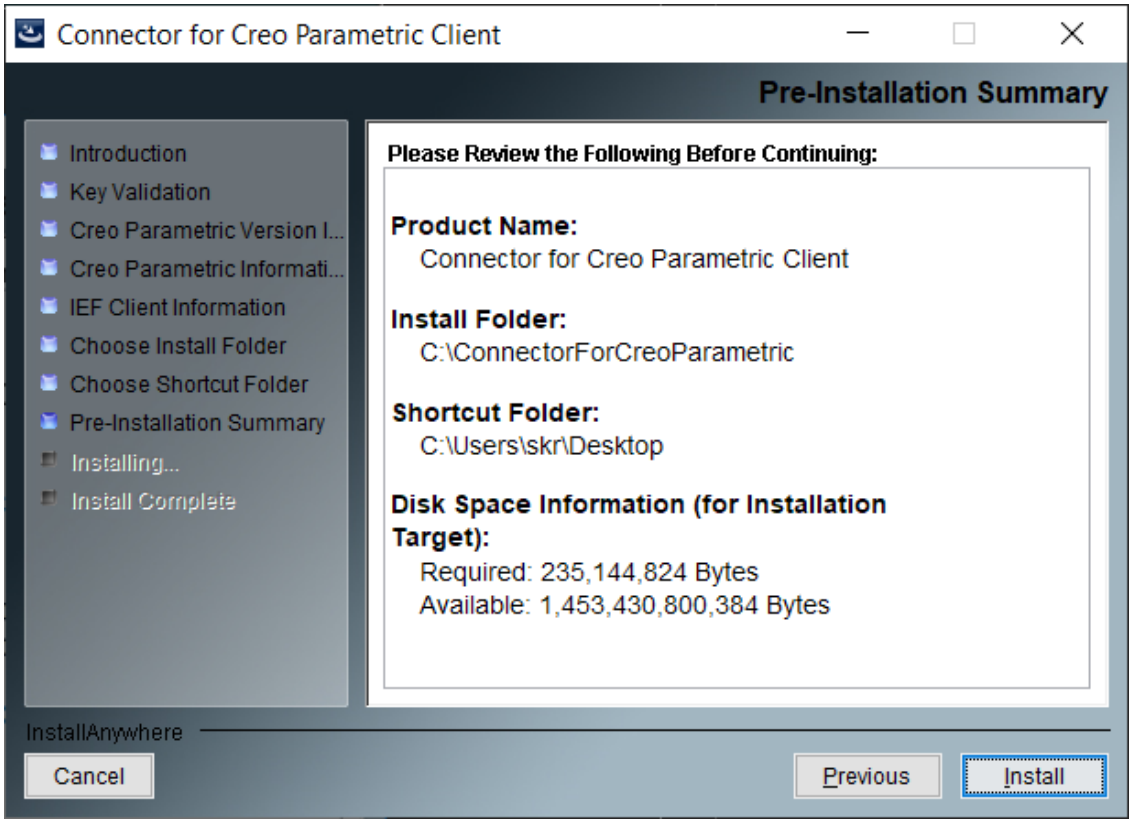
You are asked where you would like to create the Product Icons, and given various choices.



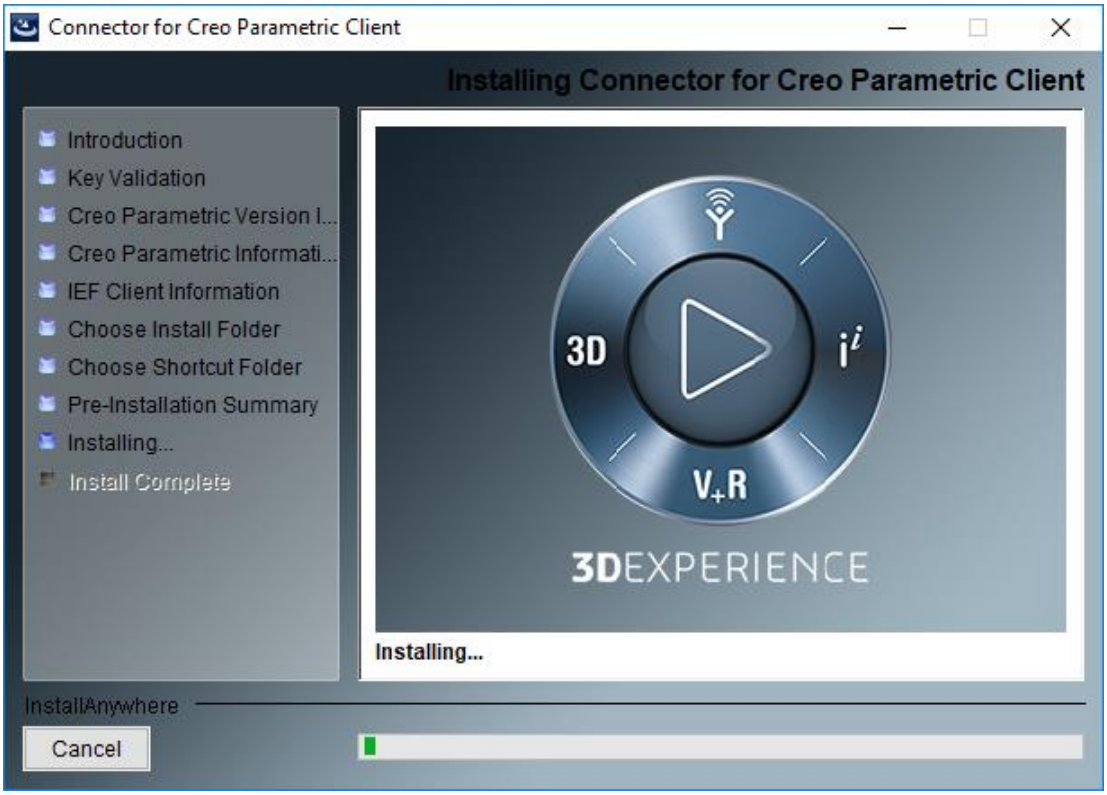
Make a selection and select **Next**

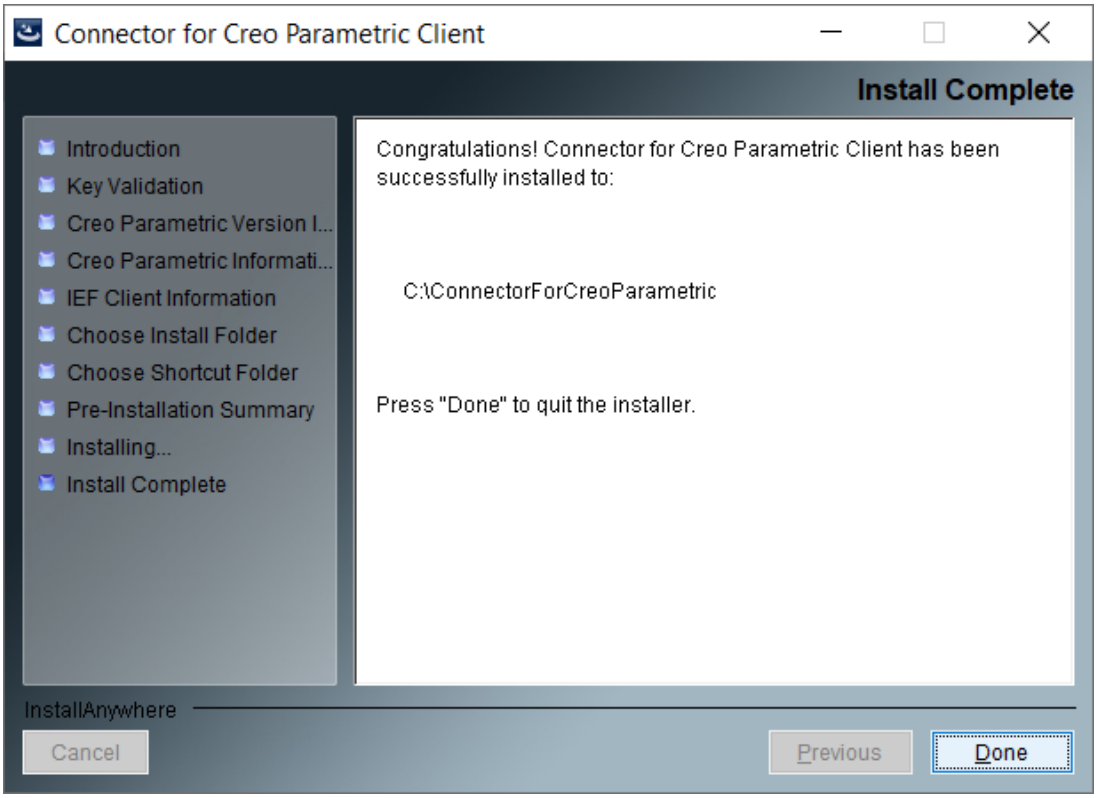


The installer will display the Pre-Installation Summary.



Select **Install**. A progress indicator will display showing the status of the install.

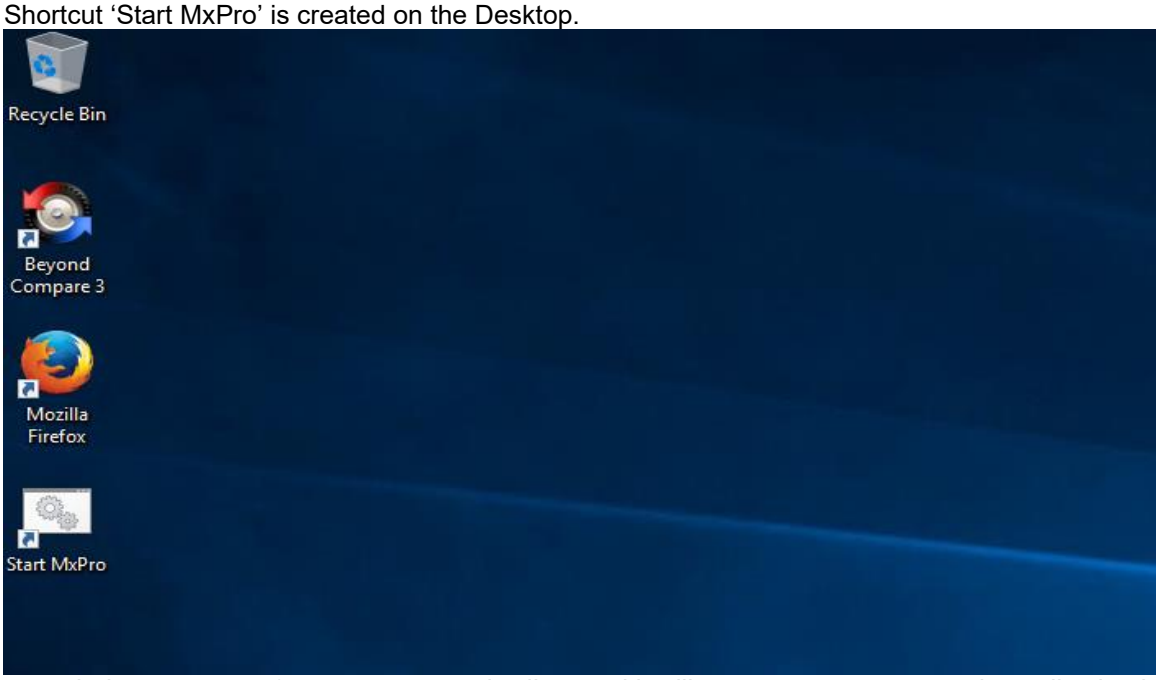




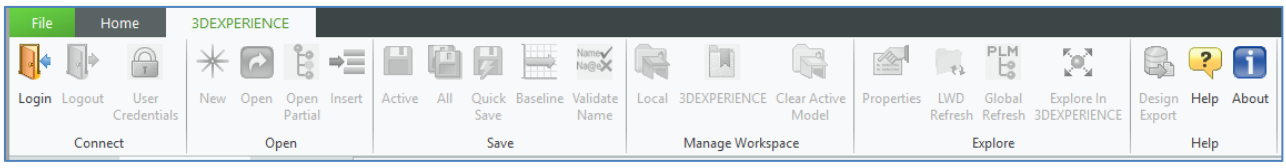
Upon completion, a window will display with information regarding the install.

Select **Done**

Review for any installation errors or warnings. The Connector for Creo Parametric Client Install is now complete. Upon successful installation of the Connector for Creo Parametric, a Desktop Shortcut ‘Start MxPro’ is created on the Desktop.



Launch the Connector for Creo Parametric client and it will open up Creo Parametric application by adding the ribbon “3DEXPERIENCE” to the Creo menu tabs.



Registering the Creo Parametric Connector

The Connector for Creo Parametric is a synchronous Pro/TOOLKIT application that requires it to be registered via a protk.dat file or with a protkdat entry in the config.pro file.

The Connector for Creo Parametric is installed with a protk.dat file that registers the Connector for Creo Parametric with Creo Parametric. This file is found in the bin directory of the Connector for Creo Parametric loadpoint. The Connector for Creo Parametric will not run if it is not properly registered with Creo Parametric. To start the Connector for Creo Parametric from a directory other than the bin directory of the Connector for Creo Parametric loadpoint, you must have a protk.dat file in your startup directory or in your home directory, or you must have a protkdat entry in your config.pro file (i.e. protkdat C:/ConnectorForCreoParametric/bin/protk.dat).

Connector for CreoParametric – Administration and Configuration

Overview

This section provides information about the administration and configuration of the Connector for Creo Parametric. It provides detailed information on how to configure various Global Configuration object attribute definitions and respective options to achieve desired behavior of the integration in the context of Creo Parametric.

The Global Configuration Object

This Connector for Creo Parametric is one of the suites of X-CAD Design-based integrations designed for use with 3DEXPERIENCE Centrals applications such as Engineering BOM. All X-CAD Design-based integrations are configurable through an object called the global configuration object. The base type is MCADInteg-GlobalConfig. Because the Connector for Creo Parametric adds attributes specific to the Connector for Creo Parametric, a subtype, MxPRO-GlobalCofig, of the global configuration object type is created. It typically has a name of ProENewArch and with revision of TEAM. The global configuration object has the following attributes.

MCADInteg-EnableHashcodeComputation: (Boolean, *Default Value: TRUE*)

This flag controls whether or not X-CAD Design will perform hash code computation comparisons on files during checkin and checkout. If this flag is set to true, X-CAD Design will create a hash code computation for the files getting checked in and store this in an attribute with the business object. This is used to determine if the file getting checked in differs from that already in 3DEXPERIENCE. X-CAD Design also uses this hash code on checkout to increase performance by not checking out a file if a file with an identical name and hash code already exists in the checkout directory.

MCADInteg-MxToCADAttribMapping: (Multiline String)

This is the 3DEXPERIENCE to Creo Parametric attribute mapping. This is used for synchronizing attributes from 3DEXPERIENCE to Creo Parametric as well as the attribute synchronization operation during checkout.

The syntax for each attribute mapping from 3DEXPERIENCE to Creo Parametric is: Matrix type,Matrixattribute|ProE type Type,ProEparameter

All user-defined Creo Parametric parameters are accessible to the Connector for Creo Parametric. In addition, the Connector for Creo Parametric can also access 3DEXPERIENCE system attributes (such as description) and user-defined attributes.

3DEXPERIENCE system attributes are denoted with a $$ prefix and postfix. The 3DEXPERIENCE description ($$description$$), owner ($$owner$$), type ($$type$$), name ($$name$$), and revision ($$revision$$) are supported in this mapping.

The names of the valid Creo Parametric and 3DEXPERIENCE types are the same as used in the MCADInteg-BusTypeMapping section. Valid names for the supported Creo Parametric types are: component, componentInstance, componentFamily, assembly, assemblyInstance, assemblyFamily, drawing, format, layout, diagram, and manufacture. The 3DEXPERIENCE types and attributes specified in the mapping must exist in 3DEXPERIENCE and the attributes must be associated to the 3DEXPERIENCE types.

An example mapping from 3DEXPERIENCE to Creo Parametric is shown below. This example maps 3DEXPERIENCE attributes to Creo Parametric parameters (MATERIAL, DEFINITION, and DESCRIPTION) for parts and assemblies, and to the Creo Parametric

parameters (DEFINITION and DESCRIPTION) for drawings. Note that Creo Parametric parameters names are always upper-case.

Example:
ProE Part,ProEMaterial|component,MATERIAL
ProE Part,ProEDefinition|component,DEFINITION
ProE Part,$$description$$|component,DESCRIPTION
ProE Assembly,ProEMaterial|assembly,MATERIAL
ProE Assembly,ProEDefinition|assembly,DEFINITION
ProE Assembly,$$description$$|assembly,DESCRIPTION
ProE Drawing,ProEDefinition|drawing,DEFINITION
ProE Drawing,$$description$$|drawing,DESCRIPTION
ProE Part Instance,$$description$$|componentInstance,DESCRIPTION
ProE Assembly Instance,$$description$$|assemblyInstance,DESCRIPTION

MCADInteg-CADToMxAttribMapping: (Multiline String)

This is the Creo Parametric to 3DEXPERIENCE attribute mapping. This is used for synchronizing attributes from Creo Parametric to 3DEXPERIENCE as well as the attribute synchronization operation during checkin.

The syntax for each attribute mapping from Creo Parametric to 3DEXPERIENCE is:
ProE type,ProEparameter|Matrix type,Matrix attribute

All user-defined Creo Parametric parameters are accessible to the Connector for Creo Parametric. In addition, the Connector for Creo Parametric can also access 3DEXPERIENCE system attributes (such as description) and user-defined attributes.

3DEXPERIENCE system attributes are denoted with a $$ prefix and postfix. Only the 3DEXPERIENCE description ($$description$$) and owner ($$owner$$) are supported in this mapping.

The names of the valid Creo Parametric and 3DEXPERIENCE types are the same as used in the MCADInteg-BusTypeMapping section. Valid names for the supported Creo Parametric types are: component, componentInstance, componentFamily, assembly, assemblyInstance, assemblyFamily, drawing, format, layout, diagram, and manufacture. The 3DEXPERIENCE types and attributes specified in the mapping must exist in 3DEXPERIENCE and the attributes must be associated to the 3DEXPERIENCE types.

An example mapping from Creo Parametric to 3DEXPERIENCE is shown below. This example maps Creo Parametric parameters (MATERIAL, DEFINITION, and DESCRIPTION) to 3DEXPERIENCE attributes for parts and assemblies, and from Creo Parametric parameters (DEFINITION and DESCRIPTION) for drawings. Note that Creo Parametric parameters names are always upper-case.

Example:
component,MATERIAL|ProEPart,ProEMaterial
component,isSheetMetal|ProEPart,ProE Sheet Metal
component,DEFINITION|ProEPart,ProEDefinition
component,DESCRIPTION|ProE Part,$$description$$
assembly,MATERIAL|ProEAssembly,ProEMaterial
assembly,DEFINITION|ProEAssembly,ProEDefinition
assembly,DESCRIPTION|ProE Assembly,$$description$$
drawing,DEFINITION|ProEDrawing,ProEDefinition
drawing,DESCRIPTION|ProE Drawing,$$description$$

Special reserved keywords exist for mapping the Creo Parametric version, build and revision to 3DEXPERIENCE. These keywords are proeVersion, proeBuild, and proeRevision and must be mapped to string attributes in 3DEXPERIENCE. proeVersion returns the version of the ProENGINEER release (e.g. '2001'). proeBuild returns the build number of the ProENGINEER release (e.g. '2001200'). proeRevision returns the version and build together (e.g. '2001/2001200'). For example:
component,proeVersion|ProEPart,PROE_VERSION
component,proeBuild|ProEPart,PROE_BUILD
component,proeRevision|ProEPart,PROE_REVISION

Mapping of material Mass properties (Density, Mass, Volume, Area, center of gravity) for Part and assemblies to 3DEXPERIENCE is shown below.

component,PRO_MP_DENSITY|ProE Part,PRO_MP_DENSITY
component,PRO_MP_MASS|ProE Part,PRO_MP_MASS
component,PRO_MP_VOLUME|ProE Part,PRO_MP_VOLUME
component,PRO_MP_AREA|ProE Part,PRO_MP_AREA
assembly,PRO_MP_DENSITY|ProE Assembly,PRO_MP_DENSITY
assembly,PRO_MP_MASS|ProE Assembly,PRO_MP_MASS
assembly,PRO_MP_VOLUME|ProE Assembly,PRO_MP_VOLUME
assembly,PRO_MP_AREA|ProE Assembly,PRO_MP_AREA

MCADInteg-RelationShipClassMapping: (Multiline String)

This attribute classifies the relationships. This attribute is generally not altered. This is a system level attribute that tells X-CAD Design how it should treat each relationship. The left side of each entry denotes the Connector for Creo Parametric - defined relationship and the right side denotes the class of the relationship.
The default value for the relationship classes are as follows

Default Value:
assemblyComponent|AssemblyLike,CADSubComponentLike
drawing|AssemblyLike
format|AssemblyLike
derived|AssemblyLike
merged|AssemblyLike,ExternalRefereneLike
mfg|AssemblyLike,ExternalRefereneLike,CircularExternalReferenceLike
mdlDependency|AssemblyLike,ExternalRefereneLike,CircularExternalReferenceLike
idf|AssemblyLike
InstanceOf|FamilyLike
DerivedOutputOf|DerivedOutputLike
ViewableOf|DerivedOutputLike

MCADInteg-TypePolicyMapping: (Multiline String)

This attribute defines the policies to use for each type. Each entry is a map of two strings separated by a separator "|". The left side of the "|" is the type and the right side is the policy that controls it. Note that the types and policies must exist in 3DEXPERIENCE.

Default Value:
ProE Part|Design TEAM Definition
ProE Part Instance|Design TEAM Definition
ProE Part Family Table|Design TEAM Definition
ProE Assembly|Design TEAM Definition
ProE Assembly Instance|Design TEAM Definition
ProE Assembly Family Table|Design TEAM Definition

ProE Drawing|Design TEAM Definition
ProE Format|Design TEAM Definition
ProE Layout|Design TEAM Definition
ProE Diagram|Design TEAM Definition
ProE Manufacture|Design TEAM Definition
ProE Note|Design TEAM Definition
IDF Model|Design TEAM Definition
Derived Output|Derived Output TEAM Policy
Viewable|Viewable TEAM Policy
CgrViewable|Viewable TEAM Policy
ThumbnailViewable|Viewable TEAM Policy

MCADInteg-RelMapping: (Multiline String)

This attribute maps Connector for Creo Parametric-defined relationships to 3DEXPERIENCE relationships. Each entry is a map of two strings separated by a separator "|". The string on the left is the Connector for Creo Parametric-defined relationship. The string on the right is the 3DEXPERIENCE relationship; a comma separated list of two entries. The second entry (e.g. CAD SubComponent) is the name of corresponding relationship in 3DEXPERIENCE. The first entry indicates the direction of arrowhead for the relationship, either to or from.

Default Value:
assemblyComponent|to, CAD SubComponent
drawing|from, Associated Drawing
format|to, ProEFormat
derived|to, ProEDerived
merged|to, ProEMerged
mfg|to, ProEManufactured
mdlDependency|to, ProEDependency
idf|from, Associated IDF
InstanceOf|to, Instance Of
DerivedOutputOf|to, Derived Output
ViewableOf|to, Viewable

MCADInteg-BusTypeMapping: (Multiline String)

This maps the business types from Creo Parametric to 3DEXPERIENCE. The types on the left side of the separator can't change but the 3DEXPERIENCE types they are mapped to can be modified.

Default Value:
assembly|ProE Assembly
assemblyInstance|ProE Assembly Instance
assemblyFamily|ProE Assembly Family Table
component|ProE Part
componentInstance|ProE Part Instance
componentFamily|ProE Part Family Table
drawing|ProE Drawing
format|ProE Format
layout|ProE Layout
diagram|ProE Diagram
manufacture|ProE Manufacture
note|ProE Note
idf|IDF Model
image_jpg|Viewable

tif|Derived Output
postscript|Derived Output
dxf|Derived Output
iges|Derived Output
step_ap203|Derived Output
cgr|CgrViewable
cgm|Derived Output
png|ThumbnailViewable
xpr|Derived Output
xas|Derived Output

MCADInteg-EnableMajorRevCreation: (Boolean, *Default Value: TRUE*)

This attribute decides whether or not the Connector for Creo Parametric should create a major revision during first checkin. If true, the Connector for Creo Parametric creates a major revision during the first checkin. If false, the Connector for Creo Parametric does not create major revision during the first checkin. It assumes that the major revision object has already been created and connects the newly created minor version object to it.

MCADInteg-TypeFormatMapping: (Multiline String)

This attribute maps the Connector for Creo Parametric type to the 3DEXPERIENCE type and format. Each entry is a map of two strings separated by a separator "|". The string on the left is the type of Connector for Creo Parametric. The string on the right is the 3DEXPERIENCE type and its format. For example, if the entry is component|ProEPart,prt, then components are checked into the prt format.

Default Value:
component|ProEPart,prt
componentInstance|ProE Part Instance,prt
componentFamily|ProE Part Family Table,prt
assembly|ProEAssembly,asm
assemblyInstance|ProE Assembly Instance,asm
assemblyFamily|ProE Assembly Family Table,asm
drawing|ProEDrawing,drw
format|ProEFormat,frm
layout|ProELayout,lay
diagram|ProEDiagram,dgm
manufacture|ProEManufacture,mfg
note|ProENote,TXT
idf|IDFModel,idf
component|ProE Part Template,prt
assembly|ProE Assembly Template,asm
drawing|ProE Drawing Template,drw
image_jpg|ProEPart,Image
image_jpg|ProE Part Instance,Image
image_jpg|ProEAssembly,Image
image_jpg|ProE Assembly Instance,Image
image_jpg|ProEDrawing,Image
image_jpg|Viewable,Image
image_jpg|Viewable,THUMBNAIL
image_jpg|ThumbnailViewable,THUMBNAIL
tif|ProEPart,TIF
tif|ProE Part Instance,TIF
tif|ProEAssembly,TIF

tif|ProE Assembly Instance,TIF
tif|DerivedOutput,TIF
postscript|ProEDrawing,PS
postscript|DerivedOutput,PS
dxf|ProEDrawing,DXF
dxf|DerivedOutput,DXF
iges|ProEPart,IGES
iges|ProE Part Instance,IGES
iges|ProEAssembly,IGES
iges|ProE Assembly Instance,IGES
iges|ProEDrawing,IGES
iges|DerivedOutput,IGES
step_ap203|ProEPart,STEP AP203
step_ap203|ProE Part Instance,STEP AP203
step_ap203|ProEAssembly,STEP AP203
step_ap203|ProE Assembly Instance,STEP AP203
step_ap203|DerivedOutput,STEP AP203
cgr|ProEPart,CGR
cgr|ProE Part Instance,CGR
cgr|ProEAssembly,CGR
cgr|ProE Assembly Instance,CGR
cgr|CgrViewable,CGR
cgm|ProEDrawing,CGM
cgm|DerivedOutput,CGM
png|ProEPart,PNG
png|ProE Part Instance,PNG
png|ProEAssembly,PNG
png|ProE Assembly Instance,PNG
png|ThumbnailViewable,THUMBNAIL
pdf|ProEDrawing,PDF
pdf|DerivedOutput,PDF
xpr|ProE Part Instance,XPR
xpr|DerivedOutput,XPR
xas|ProE Assembly Instance,XAS
xas|DerivedOutput,XAS

MCADInteg-CADToMxRelAttribMapping: (Multiline String)

This is the Creo Parametric to 3DEXPERIENCE attribute mappings for relationships. This is also used for the attribute synchronization operation during checkin but only if the users' local preference is set up to copy relationships attributes on checkin.

Default Value:
assemblyComponent,relativeXform|CADSubComponent,Spatial Location
assemblyComponent,componentID|CADSubComponent,Reference Designator

Note: Reference Designator attribute value should not be edited by user. It is meant to store unique id between assembly and component.

These two mappings must exist and the users' local preference must be set to copy relationship attributes on checkin in order to modify checked in structures in 3DEXPERIENCE.

The reserved word componentName can be used to map the name of an assembly feature component from Creo Parametric to 3DEXPERIENCE. For example:

assemblyComponent,componentName|CADSubComponent,Component Name

MCADInteg-MxToCADRelAttribMapping: (Multiline String)

This is the 3DEXPERIENCE to Creo Parametric attribute mappings for relationships. This is also used for the attribute synchronization operation during checkout but only if the user's local preference is set up to copy relationships attributes on checkout.

Default Value:
CAD SubComponent,SpatialLocation|assemblyComponent,relativeXform
CAD SubComponent,ReferenceDesignator|assemblyComponent,componentID

Note: Reference Designator attribute value should not be edited by user. It is meant to store unique id between assembly and component.

The reserved word componentName can be used to update the name of an assembly feature component in Creo Parametric from 3DEXPERIENCE. For example:
CAD SubComponent,ComponentName|assemblyComponent,componentName

MCADInteg-ExpandedSubComponent: (Boolean, *Default Value: TRUE*)

This flag controls the rolling up behavior for "CAD SubComponent" relationship. If there are shared components in an assembly and if this flag is set to TRUE multiple "CAD SubComponent" relationships are created between the shared component and the assembly. If this attribute is set to FALSE only one "CAD SubComponent" relationship is created between the shared component and the assembly and the "Quantity" attribute on "CAD SubComponent" relationship is set to the number of occurrences of the shared component.

MCADInteg-NonSupportedCharacters: (Multiline String)

This attribute holds the characters that are not supported in file names. If any Creo Parametric file getting checked in has any of the characters listed in this attribute a message is shown to the user and checkin is not allowed.

Default Value:
@*&?#%${}()<>|/.,;:\

MCADInteg-CreateDerivedOutputObj: (Boolean, *Deprecated*)

This flag is deprecated by xCAD Design. Now the behavior is to always create separate derived output objects and it cannot be turned off.

MCADInteg-TypeDerivedOutputMapping: (Multiline String)

This attribute contains the mapping between business types and Derived Output types. This mapping determines the type of the Derived Output object that is to be created (if MCADInteg-CreateDerivedOutputObj is true) for mapped business types. Each mapping contains two entries separated by a "|". The entry on the left is a business type in the Connector for Creo Parametric context and the entry on the right contains Derived Output type(s). If there are multiple Derived Output types they must be separated by commas.

Default Value:
assembly|image_jpg,tif,iges,step_ap203,cgr,png
component|image_jpg,tif,iges,step_ap203,cgr,png
drawing|image_jpg,postscript,dxf,iges,cgm,pdf

assemblyInstance|image_jpg,tif,iges,step_ap203,cgr,png,xas
componentInstance|image_jpg,tif,iges,step_ap203,cgr,png,xpr
format|
layout|
diagram|

MCADInteg-FeatureJPOMapping: (Multiline String)

This attribute contains the mapping of the feature and the name of the JPO that is to be invoked for that feature.

Default Value:
EBOMSynchronize|MCAD-EBOMSynchronize
Finalize|MCAD-Finalize
UndoFinalize|MCAD-UndoFinalize
Purge|MCAD-Purge
Rename|MCAD-Rename
SaveAs|MCAD-SaveAs

The string on the left represents the operation, like finalize, EBOM synchronize etc. The value on the right is the name of program, a JPO that gets invoked that has all the logic for the operation implemented. All the programs indicated in the above setting are installed by the out-of-the box installation.

MCADInteg-TypeClassMapping: (Multiline String)

This attribute defines the class for business types. This is a system level attribute and this attribute should not be modified. This attribute has two entries separated by a "|". The entry on the left defines the class for the business types, while the entry on the right contains a comma-separated list of Connector for Creo Parametric types.

Default Value:
TYPE_CADMODEL_LIKE|assembly,component,drawing,assemblyFamily,componentFamily,layout,format,diagram,manufacture
TYPE_COMPONENT_LIKE|component,componentInstance
TYPE_ASSEMBLY_LIKE|assembly,assemblyInstance
TYPE_FAMILY_LIKE|assemblyFamily,componentFamily
TYPE_INSTANCE_LIKE|assemblyInstance,componentInstance
TYPE_DERIVEDOUTPUT_LIKE|image_jpg,tif,postscript,dxf,iges,step_ap203,cgr,cgm,png,pdf,xpr,xas
TYPE_COMPONENT_FAMILY_LIKE|componentFamily
TYPE_ASSEMBLY_FAMILY_LIKE|assemblyFamily

MCADInteg-RenameFilesInServer: (Boolean, *Default Value: FALSE*)

This flag tells the Connector for Creo Parametric whether the files are to be renamed on the server side during the Rename and SaveAs operations. If this flag is set to true then the files are renamed on the server side during these operations. If the flag is set to false then no file renaming operation is done on the server side during Rename and SaveAs. This must be set to FALSE for this Connector for Creo Parametric and must not be changed.

IEF-CSEDisabledCommandsList (String):

This attribute is used to disable ProE commands. Add command names in form of comma-separated list.

Command names for disabled commands:
New,Open,OpenPartial,Insert,SaveActive,SaveAll,QuickSave,Baseline,GlobalRefresh,Properties,WorkspaceLocal,Workspace3DEXPERIENCE,WorkspaceClearActiveModel,ExploreIn3DEXPERIENCE,PLMStatus,Help,About

Creo Parametric integration menu command ‘OpenPartial’ is disabled by default as it has dependencies on the Server side schema that is not installed by Connector for Creo Parametric.

IEF-PreferencesPageLayoutJPO (String):

This attribute holds the name of the program (JPO) to be invoked for creating the page layout for the Preference page.

Default Value:
IEFPreferencesPageLayout

IEF-TypeTemplateMapping (Multiline String):

This attribute holds the mapping of cad types to template types required for start design. The format for this mapping is:

CAD_TYPE|TEMPLATE_TYPE

Default Value:
component|ProE Part Template
assembly|ProE Assembly Template
note|ProE Note Template
drawing|ProE Drawing Template

IEF-EBOMSync-RegistryTNR (String):

Pipe delimited type|name|revision of the EBOM Synch configuration object.

Default Value:
MCAD-EBOMSyncConfig|MCAD-EBOMSyncConfig|-

IEF-SourceDetails (String):

This attribute holds the name of the Connector for Creo Parametric and is currently being used for the start design feature. The CSE version may optionally be provided by appending a "|" to the Connector for Creo Parametric name. The format for this mapping is:

CSE_NAME[|CSE_VERSION]

Default Value:
MxPRO

IEF-CADToMxMandatoryAttrMapping: (Multiline String)

This attribute defines the mapping of mandatory attributes from ProE to 3DEXPERIENCE. Attribute mappings in this section are, by definition, required to be copied from ProE to 3DEXPERIENCE on checkin. Attribute mappings in this section must also be defined in the MCADInteg-CADToMxAttribMapping section.

Default Value:
#CADType|Enovia Attribute Name|Default Value

The default value shows the format in which the entry to this GCO setting is to be made:

- CADType: The Creo Parametric Type supported in 3DEXPERIENCE like component, assembly, componentInstance, assemblyInstance, drawing.
- Enovia Attribute name: 3DEXPERIENCE attribute name associated with corresponding 3DEXPERIENCE ProE types
- Default Value: This is an optional entry, you can assign a default value for mandatory attribute.

Example:
#CADType|Enovia Attribute Name|Default Value
component|$$description$$|HEADER, PCB, FRICTION LOCK
component|WEIGHT|0.0
component|MATERIAL|04001234(PP,Black)
assembly|$$description$$|
assembly|WEIGHT|
drawing|$$description$$|

ProEAutoLoadMasterRep: (String, *Default Value: FirstTime*)

This is an attribute specific to the Connector for Creo Parametric. Possible values are:

- FirstTime
- Always
- Never

When a ProE assembly with a simplified representation active is being saved to 3DEXPERIENCE this attribute controls whether or not the Master representation is automatically loaded prior to the save operation.

When this attribute is set to FirstTime, the default, the Master representation will be automatically loaded if the assembly does not already exist in the 3DEXPERIENCE database.

When this attribute is set to Always, the Master representation will always be automatically loaded prior to the save operation, regardless of whether or not the assembly has been previously saved to the 3DEXPERIENCE database.

When this attribute is set to Never, the Master representation will never be automatically loaded prior to the save operation, regardless of whether or not the assembly has been previously saved to the 3DEXPERIENCE database.

Setting the ProEAutoLoadMasterRep attribute to FirstTime or Never can provide improved performance by not having to activate the entire Master representation prior to every save of the assembly. However, the user must take care to ensure that all components participating in the Master and other simplified reps are saved to the 3DEXPERIENCE database. In general, this can be ensured by:

- If the assembly is modified to include a new component, then the assembly should be checked in with the master rep (or a simplified rep that includes that new component) active.

Note: If a representation of an assembly, master or specific representation, is opened via 3DEXPERIENCE/Open but it has missing components (e.g., the assembly was never saved with the master or specific representation active), the requested representation of the assembly will be opened by suppressing the missing components. Messages regarding the missing components will be written to the text area of Creo Parametric. If the assembly is to be saved to 3DEXPERIENCE again, care must be taken to retrieve missing components and to unsuppress them in the assembly.

ProEDerivedOutputOptions: (Multiline String)

This is an attribute specific to the Connector for Creo Parametric. This attribute defines the user-configurable options of image file and derived output generation. The syntax is:

derivedoutput|option=value,<option=value,…>

Default Value:
image_jpg|width=5.0,height=5.0,depth=8,dpi=100,delete_on_checkin=yes
tif|width=5.0,height=5.0,depth=8,dpi=100,delete_on_checkin=yes
postscript|size=auto,variable_width=0.0,variable_height=0.0,quality=3,scale=1,pen_slew=no,pen_velocity_x=0.0,pen_velocity_y=0.0,pages=all,first_page=0,last_page=0,segmented_output=no,label=no,separate_files=no,delete_on_checkin=yes
cgr|chord_height=0.1,angle_control=0.5,delete_on_checkin=yes
cgm|file_structure=cleartext,coordinates=abstract,all_sheets=yes,delete_on_checkin=yes
dxf|all_sheets=yes,delete_on_checkin=yes
iges|delete_on_checkin=yes
step_ap203|delete_on_checkin=yes
png|width=400,height=600,delete_on_checkin=yes
pdf|depth=color,dpi=300,pages=all,delete_on_checkin=yes

Note:
delete_on_checkin specifies whether or not to delete the derived output file upon successful completion of the checkin operation. If this is set to ‘yes’ the derived output file will be deleted upon checkin. If this is set to ‘no’ the derived output file will not be deleted.

ProEMissingLinksWarning : (Boolean, *Default Value: FALSE*)

This is an attribute specific to the Connector for Creo Parametric. The Connector for Creo Parametric will notify the user of missing links corresponding to non-required relationships when checkin is selected from the 3DEXPERIENCE menu. The user will be able to continue or cancel the checkin operation. If the user chooses to continue the checkin operation the Connector for Creo Parametric can either prevent those parent models with missing links from being selected for checkin or can ignore the missing links and allow the parent models to be selected for checkin. This behavior is controlled by the GCO attribute ProEUnloadedChildRelationshipExceptions.

ProEObjectLoadingInSession: (String, *Default Value: LoadObjectsInSession*)

This is an attribute specific to the Connector for Creo Parametric. Possible values are 'LoadObjectsInSession' and 'DoNotLoadObjectsInSession'. This attribute defines whether or not to load objects that aren't in session. This affects non-required related items that are not in session. For example, layouts may have references to many other objects that aren't loaded in session. If this attribute is set to 'DoNotLoadObjectsInSession' the Connector for Creo Parametric will not attempt to bring the unloaded objects into session. When this option is set to 'DoNotLoadObjectsInSession', derived outputs will not be generated for the objects which have not been loaded.

ProEPartFamilyInstanceHandling: (String, *Default Value: PartFamilyInstanceLight*)

This is an attribute specific to the Design TEAM Definition Managment. Possible values are 'PartFamilyInstanceLight' and 'PartFamilyInstanceLoaded'. This attribute defines whether or not part family instances are generated and retrieved into session during processing of the checkin command (and lock/unlock). When the option is set to PartFamilyInstanceLoaded, each part family instance is retrieved into session to interrogate the structure, check for nested instances, and generate derived outputs. When the option is set to 'PartFamilyInstanceLight' the instances are read from the ProE model file without loading each instance into session. 'PartFamilyInstanceLight' is a significant performance gain over 'PartFamilyInstanceLoaded'. The limitation is that some operations (such as derived output generation) can not be performed on instances that are already in session. When using the 'PartFamilyInstanceLight' setting, it is possible for the user to generate derived outputs by loading specific instances into session.

ProEAssemblyFamilyInstanceHandling: (String,D*efaultValue:AssemblyFamilyInstanceLight*)

This is an attribute specific to the Connector for Creo Parametric. Possible values are 'AssemblyFamilyInstanceLight' and 'AssemblyFamilyInstanceLoaded'. This attribute defines whether or not assembly family instances are generated and retrieved into session during processing of the checkin command (and lock/unlock). When the option is set to 'AssemblyFamilyInstanceLoaded', each assembly family instance is retrieved into session and generates derived outputs. When the option is set to 'AssemblyFamilyInstanceLight' 'the instances are read from the ProE model file without loading each assembly instance into session. 'AssemblyFamilyInstanceLight' is a significant performance gain over 'AssemblyFamilyInstanceLoaded'.

ProEUnloadedChildRelationshipExceptions: (String, *Default Value: ''*)

This is an attribute specific to the Connector for Creo Parametric. This allows a comma-separated list of relationship names as found on the left hand side of the mappings in the MCADInteg-RelMapping GCO attribute (e.g. assemblyComponent, drawing, format, derived, merged, mfg, mdlDependency). The default value is an empty string. The purpose of this setting is to ignore missing children depending on their relationship to the parent model. The Connector for Creo Parametric does not allow any model to be selectable for checkin if any of its related children are missing, in order to avoid checking in incomplete structures. This can happen for non-required related children that aren't necessary for successful regeneration of the parent model. It may be desirable to make exceptions for certain types of relationships where it may be common for non-required related children to

be missing on disk. This most often occurs with part-to-part or part-to-assembly relationships which are mapped as ‘merged’ and ‘mdlDependency’ relationships.

If this value is empty, no exceptions will be made and any missing child will result in the parent object not selectable for checkin.

If this value is set to mdlDependency,merged then an exception will be made for any missing child of a mdlDependency or merged relationship and the parent object will be selectable for checkin.

MCADInteg-UpdateAttribMapping: (Multiline String)

This is an attribute specific to the Connector for Creo Parametric. Using the Attribute Update feature and this GCO setting, an integration user can add multiple attributes or parameters to the in-session models. The user would see all the mapped parameters present in this GCO option in the Mass Attribute dialog, which can be invoked from the Save dialog.

Default Value:

1|DESCRIPTION|$$description$$

The format consists of three fields:

- The first entry: determine the column sequence in Attribute Update UI during checkin operation.
- The second entry: CAD parameter name,
- The third entry: corresponding 3DEXPERIENCE attribute name.

Example:
1|DESCRIPTION|$$description$$
2|MATERIAL|MATERIAL
3|WEIGHT|WEIGHT

MCADInteg-ForceWorkspaceOnSave: (Boolean, *Default Value: TRUE*)

This is an attribute specific to the Connector for Creo Parametric. This GCO setting enforces users to select a workspace folder in the Save dialog.

ProEAlternateTNRSource: (Boolean, *Default Value: FALSE*)

This is an attribute specific to the Connector for Creo Parametric to specify if the Connector for Creo Parametric should use an alternate TNR source for auto-recognition of models in 3DEXPERIENCE. The purpose of the alternate TNR source is for models migrated to 3DEXPERIENCE from Pro/PDM. If auto-recognition is active, on checkin if the TNR stamping is not found in the model file and this attribute is set then the Connector for Creo Parametric will use the PDMRL and PDMDB reserved parameters in the model files.

These parameters will have to be populated after a migration effort in order to be used. PDMRL must have the 3DEXPERIENCE type and revision in the format TYPE|REV, for example PEVA|A.0. The type will be an abbreviation of the actual 3DEXPERIENCE type. The full set of abbreviations is:

ProE Versioned Assembly = PEVA
ProE Versioned Assembly Family Table = PEVAFT
ProE Versioned Assembly Instance = PEVAI
ProE Versioned Diagram = PEVDIA
ProE Versioned Drawing = PEVD

ProE Versioned Format = PEVF
ProE Versioned Layout = PEVL
ProE Versioned Manufacture = PEVM
ProE Versioned Part = PEVP
ProE Versioned Part Family Table = PEVPFT
ProE Versioned Part Instance = PEVPI

PDMDB must have the 3DEXPERIENCE object name which must always be upper case. This should be identical to the ProE model name.

This alternate TNR source will not be used again after a checkin following a migration because the Connector for Creo Parametric will stamp and use the new TNR in the default manner.

MCADInteg-ENOVIANewAttributeMapping

This attribute is used for new design creation in Integration. In addition to this it provides the mapping for EC Part Number and its attributes. While creating a new Design object, the mapped attributes are added to CAD file as its parameters.
The syntax for mapping attribute to Creo Parameter object is:
\[START CAD ATTRIBUTES\]
CAD Type|PARAMETERNAME=Attribute Name{Attribute Type~Mandatory(Y/N)~DefaultValue}
Multiple PARAMETERNAME settings for given CAD type will be appended with '@' symbol.
\[END CAD ATTRIBUTES\]
\[START ECPART ATTRIBUTES\]
Part Type|Attribute Name{Attribute Type~Mandatory(Y/N)~DefaultValue}
\[END ECPART ATTRIBUTES\]

The format consists of following fields:

- CAD type: CAD type from 3DEXPERIENCE
- PARAMETERNAME: Name of the CAD attribute to be created on new design
- Attribute Name: This is optional value for setting CAD attribute. This is used as label for the Creo Parametric attribute to be used in integration UI during new design creation.
- Attribute type: This entry refers to 3DEXPERENCE attribute type.
- Mandatory(Y/N): If set to 'Y', value of the attribute value cannot be blank
    - If set to 'N', value of the attribute value may or may not be blank.
- DefaultValue: the attribute can be assigned with a default value.
- Part Type: This entry is made for the EC Part attribute. Part Type can be a specific type such as Part, or the Common needs to be configured for the attributes that are common to all Part Types. Common is applicable only for EC Parts

Example:

[START CAD ATTRIBUTES]
component|DESCRIPTION=Description{string~N}@MATERIAL{string~N~04001234(PP,Black);04001640(PTFE)}
componentInstance|DESCRIPTION=Description{string~N}@MATERIAL{string~N~04001234(PP,Black);04001640(PTFE)}
assembly|DESCRIPTION=Description{string~N}@MATERIAL{string~N~04001234(PP,Black);04001640(PTFE)}
drawing|DESCRIPTION=Description{string~N}
[END CAD ATTRIBUTES]
[START ECPART ATTRIBUTES]
Common|
Part|
[END ECPART ATTRIBUTES]

Establish Attribute Mappings

The Global Configuration Object can be configured for Creo Parametric parameters to be mapped to 3DEXPERIENCE attributes and for 3DEXPERIENCE attributes to be mapped to Creo Parametric parameters. The MCADInteg-MxToCADAttribMapping section establishes the mappings from 3DEXPERIENCE to Creo Parametric and the MCADInteg-CADToMxAttribMapping section establishes the mappings from Creo Parametric to 3DEXPERIENCE.

Refer to the MCADInteg-MxToCADAttribMapping and MCADInteg-CADToMxAttribMapping attributes of The Global Configuration Object section for a complete explanation of establishing attribute mappings.

Adding a New Type

Additional types can be created and used by X-CAD Design-based integrations. Here are the steps required to create and use a new type. This example will show how to create a new type called ProE Special Part.

- **Define the new type**
  In MQL, create a new type. Derive that from the appropriate type per your requirements.

  Example:
  Create 'ProE Special Part' derived from 'ProE Part' or 'MCAD Component'.

- **Add the new types to their governing policies**
  In MQL add these two new types to the list of governed types for their governing policies. New types can either be added to existing policies or new policies can be created to govern the new types. The default policy for all types is 'Design TEAM Definition'

  Example:
  Add 'ProE Special Part' to the 'Design TEAM Definition' policy. Note that if 'ProE Special Part' is derived from the 'CAD Model' or a type that is derived from the 'CAD Model' type then you don't need to add 'ProE Special Part' to the 'Design TEAM Definition' policy because the 'CAD Model' type is already governed by this policy.

- **Modify MCADInteg-BusTypeMapping**
  In 3DEXPERIENCE add a mapping for the new type to the MCADInteg-BusTypeMapping attribute of the Global Configuration Object. Replace the old bus type mapping with new bus type.

  Example:
  component|ProE Special Part

- **Modify MCADInteg-TypePolicyMapping**
  In 3DEXPERIENCE add a mapping for the policy of the new type to the MCADInteg-TypePolicyMapping attribute of the Global Configuration Object. Mappings need to be added for both of the new types.

  Example:
  ProE Special Part|Design TEAM Definition

- **Modify the MCADInteg-TypeFormatMapping**
  In 3DEXPERIENCE add a mapping for the format of the new type to the MCADInteg-TypeFormatMapping attribute of the Global Configuration Object. Do not add a mapping for the new versioned type.

  Example:
  component|ProE Part, ProE Special Part,prt

- **Modify the IEF-Pref-MCADInteg-DefaultTypePolicySettings**
  In 3DEXPERIENCE add a mapping for the default policy of the new type in the IEF-Pref-MCADInteg-DefaultTypePolicySettings attribute of the Global Configuration Object.

  Example:
  (DEFAULTVALUE)ProE Special Part|Design TEAM Definition

- **Modify the trigger MCADDeleteFiles_if.tcl**
  In MQL modify the program MCADDeleteFiles_if.tcl. Add the new major type (not the versioned type) to the list LtypeFormatList which is initialized near the beginning of the program. The format of this list is explained in the comments above the list. This program is used by X-CAD Design and the major type (and the formats it uses) must be added to this list for X-CAD Design to properly function.

- **Register the new types with 3DEXPERIENCE**
  In 3DEXPERIENCE you need to register the new types in order for the web client type chooser to include them.

Derived Output Support

The Connector for Creo Parametric supports the checkin of derived output files. These derived output files can be dynamically generated (native) or may already exist on disk (user-defined).

Native derived outputs

The Connector for Creo Parametric uses the native CAD tool API's to programmatically generate these file during checking based on the users' preferences and selections. The files are generated in the same directory as the ProE files and are deleted upon completion of the checkin depending on the value of delete_on_checkin in the ProEDerivedOutputOptions. The Connector for Creo Parametric supports the following native derived outputs with the necessary additions to the global configuration object.

Parts and assemblies:
JPEG
TIF
IGES
STEP_AP203
CGR
PNG

Drawings:
Image_jpg
POSTSCRIPT
DXF
IGES
CGM
PDF

For ProE Part Instances, the integration supports XPR derived output in addition to the derived outputs mentioned under that "Parts and Assemblies. Similarly, in addition to the derived outputs mentioned under the Parts and Assemblies, the integration supports XAS derived output for ProE Assembly Instances.

ProE allows some level of control over the generation of JPEG, TIF, and POSTSCRIPT files. Refer to the ProEDerivedOutputOptions section for more information.

Refer to the CGR Derived Output section for more information on the CGR derived output.

User-defined derived outputs

The Connector for Creo Parametric checks for user-defined derived output files in the same directory as the ProE files and will check them in if found. The user-defined file must have the same basename as the CAD file being checked in. The files are deleted upon completion of the checkin.

User-defined derived outputs are identified with the syntax of:
userfile_<extension>

For example, to support zip files the name would be userfile_zip.

plot_* derived outputs

The Connector for Creo Parametric also has a mechanism to generate plot files for supported ProE plotters. These are defined with a prefix of 'plot_' followed by the plotter

name (plot_<plotter name>). For example, if you define plot_HP7585B as a derived output, the Connector for Creo Parametric will generate a plot file for the HP7585B plotter.

The user-defined derived outputs must be added to the global configuration object. The MCADInteg-BusTypeMapping, MCADInteg-TypeClassMapping, MCADInteg-TypeDerivedOutputMapping, and MCADInteg-TypeFormatMapping attributes of the global configuration object must be edited. The chosen format may also need to be added to 3DEXPERIENCE and to the allowed formats of the governing policies, and the range for the 'CAD Type' attribute will have to be updated.

All the derived outputs will be checked in separate formats of the same Derived Output Object or the CAD Type Object depending on the user setting as follows:

- If the MCADInteg-CreateDerivedOutoutObj flag in the GCO is set to TRUE, the derived outputs will be checked in the corresponding formats of the same Derived Output object linked to the CAD Model.
- If the MCADInteg-CreateDerivedOutoutObj flag in the GCO is set to FALSE, the derived outputs will be checked in the corresponding formats of the same CAD Model object.

CGR Derived Output for Downstream Applications

For users that are generating CATIA Graphics Representation (CGR) files as optional derived output for use in downstream applications, there are 3DEXPERIENCE parameters that must be established and decisions on the refinement required of the CGR files.

Global Configuration Object settings

To use CGR files in other 3DEXPERIENCE Applications the following Global Configuration Object(GCO) parameters must be set before attempted CGR creation during a Save operation. The Integration Administrator privileges are required to modify the GCO in the thick client user interface.

MxPRO-GlobalConfig "ProENewArch"	Required value
MCADInteg-CADToMxRelAttribMapping	assemblyComponent,relativeXform|CADSubComponent,Spatial Location
IEF-Pref-MCADInteg-SelectedDerivedOutput	(DEFAULTVALUE)cgrOutput
ProEDerivedOutputOptions	cgr|chord_height=0.1,angle_control=0.5,delete_on_checkin=yes

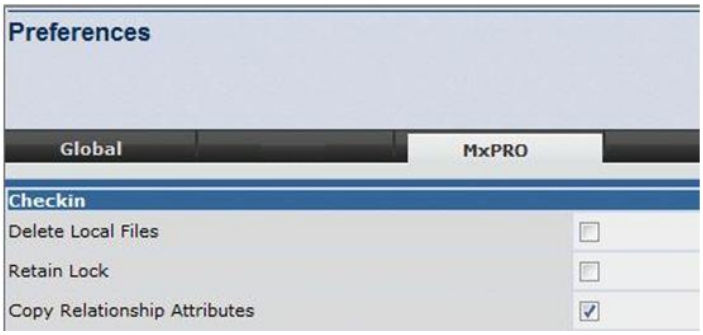
ProEDerivedOutputOptions

It should be noted that these variables directly control the accuracy of the tessellation of the native Creo Parametric file for subsequent creation of the CGR file. The tighter the values the greater accuracy of the CGR file, however, as the tolerances are increased the time required for calculating the CGR and subsequent storage in 3DEXPERIENCE also increases.

The UNITS of the GCO variable "ProEDerivedOutputOptions" are in MM regardless of Creo Parametric session units. Additionally the CGR generated and stored in 3DEXPERIENCE is also in MM units.

Preferences Settings

In addition to the attributes above, the Preference setting of "Copy Relationship Attributes" available from within the WEB browser must be checked in the 'Checkin' section as shown below.



Adding a New Derived Output

Connector for Creo Parametric allows derived output files to be generated at the time of checkin. The primary use of this function is to create image files during checkin. The Connector for Creo Parametric provides out of the box support for TIF, JPEG, POSTSCRIPT, DXF, IGES, CGR, PNG and STEP AP203. The Connector for Creo Parametric also supports plotters that have been setup and are available in a customer's ProE environment. The File/Print menu in ProE shows the available plotters. There is also a default plotter in ProE called 'default' that can be used even though it is not seen from the File/Print menu. This example will illustrate how to modify the Connector for Creo Parametric by adding a new derived output for the 'default' ProE plotter. The name of the derived output must be 'plot_' plus the name of the plotter. For this example, the name that must be used is plot_default. If the wrong name is used or ProE doesn't recognize the name of the plotter then no file will be generated.

New derived outputs are added to the Connector for Creo Parametric by modifying the global configuration object for the Connector for Creo Parametric. The default type, name, and revision of the global configuration object is: MxPRO-GlobalConfig, ProENewArch, TEAM. Edit the global configuration object and make the following changes.

- **Modify MCADInteg-TypeClassMapping**
  Add the new derived output name to TYPE_DERIVEDOUTPUT_LIKE.

  Example:
  TYPE_DERIVEDOUTPUT_LIKE|image_jpg,image_tif,postscript,dxf,iges,step_ap203,plot_default

- **Modify MCADInteg-TypeDerivedOutputMapping**
  Add the new derived output name to the object type(s) for which it is applicable. Plotters are only valid for drawing types.

  Example:
  drawing|postscript,dxf,iges,plot_default

- **Modify MCADInteg-BusTypeMapping**
  Map the new derived output to the business type Derived Output.

  Example:
  plot_default|Derived Output

- **Modify MCADInteg-TypeFormatMapping**
  Add format mappings for the new derived output. You may have to create a new format type if you want the generated files to be checked into a format that doesn't already exist. If you create a new format you must add it to the policy that controls the ProE business object and the Derived Output business object.

  Example:
  plot_default|ProEDrawing,hpgl
  plot_default|DerivedOutput,hpgl

  (This example assumes a format called hpgl already exists in 3DEXPERIENCE and is a valid format for the policies that control ProE Drawing and Derived Output.)

- **Modify MCADInteg-CreateDerivedOutputObj**
  Set MCADInteg-CreateDerivedOutputObj to TRUE if you want to create a separate business object for the generated file. If this is set to FALSE the generated file will be checked into the same business object the native ProE file is checked into but into the format as defined in the MCADInteg-TypeFormatMapping.

- **Modify ProEDerivedOutputOptions**
  Add an entry to this section like the postscript entry. There are various settings that control the generation of the plot file. Plot files use the same settings that the postscript generation does.

  Example:
  plot_default|size=auto,variable_width=0.0,variable_height=0.0,quality=3,scale=1,pen_slew=no,pen_velocity_x=0.0,pen_velocity_y=0.0,pages=all,first_page=0,last_page=0,segmented_output=no,label=no,separate_files=no,file_extension=.plt,delete_on_checkin=yes

  size is the drawing size. Valid values are auto, A, B, C, D, E, F, A4, A3, A2, A1, A0, VARIABLE_INCH, or VARIABLE_MM. If size=auto the Connector for Creo Parametric will automatically set the size of the plot file to the size of the drawing sheet when separate_files=yes but it will set the size of the plot file to the size of the first sheet of the drawing if separate_files=no.

  variable_width and variable_height specify the width and height if size is set to VARIABLE_INCH or VARIABLE_MM.

  quality specifies the plot quality. Valid values are 1, 2, or 3.

  scale is the drawing scale. It must be a positive non-zero number.

  pen_slew must be no or yes.

  pen_velocity_x and pen_velocity_y specify the pen velocity in the x and y directions.

  pages identifies which sheets to plot. Valid values are all, current, or range.

  first_page and last_page specify the first and last sheets to plot if pages is set to range.

  segmented_output must be yes or no. It can be yes only if plotting a single sheet.

  label must be yes or no. It specifies whether or not to put a label on the plot.

  separate_files must be yes or no. If no, a single file is created. If yes, a plot file is created for each sheet.

  file_extension specifies the file extension of the generated plot file. If file_extension is not specified the extension of the plot file will be.plt.

  delete_on_checkin specifies whether or not to delete the derived output file upon successful completion of the checkin operation. If this is set to ‘yes’ the derived output file will be deleted upon checkin. If this is set to ‘no’ the derived output file will not be deleted.

You will now be able to select this new derived output for drawings. You can change your preferences if you wish to always have it pre-selected.

Rapid File Access

To successfully enable rapid file access for Creo Parametric on Windows, the shared drive must be mapped to a drive letter.

The IEF-GetHardlinkServerDetailsForRFA program in 3DEXPERIENCE must be modified by setting a “MappedDrive” property.

For example:

```
HashtabledefaultLocation = new Hashtable();
defaultLocation.put("HardlinkServerURL", "http://yourserver:7001/ematrix");
defaultLocation.put("MappedDrive", "H:");
locationHardlinkServerDetailsMapping.put("DefaultLocation", defaultLocation);
```

In addition, the hardlink folder one level above the user's hardlink folder must be shared.

CAD Classification Schema objects

A new functionality is available which allows Creo users to classify the objects right inside the CAD session using integration's 3DEXPERIENCE Save UI. Please see user guide for more details of the capability. To enable this functionality some server side components are also delivered which will be installed by the main integration schema installer. Can only be used if Library Central is already installed.
Following are the classification schema objects.

- A new JPO called "ITIClassificationAssignment" is installed. This JPO performs all the classification assignment in 3DEXPERIENCE and also communicates with the integration client for the functionality to work properly.

- An admin object is installed to manage the Classification Assignment
  MCAD Classification Assignment Control object. This will help administrators to enforce some of the classification assignment behavior.

  Type: MCAD Classification Assignment
  Name: PRO Classification Assignment Control

  One attribute for defining classification types
  Attribute1: MCADInteg-ClassificationType-Configuration
  Syntax: Classification Type|One or Many Classification Types|Mandatory(Y/N)|List of Classification objects

  For example, user is allowed to assign one Export Control Class mandatorily and optionally assign many IP Control Class
  Export Control Class|One|Y|EAR~ITAR~Not Export Controlled
  IP Control Class|Many|N|Class I~Class II~Class III Generator

  In this next example, user is allowed to assign up to 3 Export Control Class optionally by picking any one from the full list of Export Control Class objects and mandatorily pick any one of the configured classes
  Export Control Class|3|N|All
  IP Control Class|One|Y|Class I~Class II~Class III Generator

  Another attribute for defining attribute configurations for classification objects.
  Attribute2: MCADInteg-ClassificationAttribute-Configuration
  Syntax: ClassificationObjectName|AttributeName|Attribute Order in Excel Sheet|Clasification Default Value or Enovia Attribute Defualt Value|MANDATORY|

See examples in following table.

MCADInteg-ClassificationAttribute-Configuration	#ClassificationObjectName|AttributeName|Attribute Order in Excel Sheet|Classification Default Value or Enovia Attribute Defualt Value #If 4th configuration is equals to string ENOVIAATRDEFAULT - default value is retrived from attibute definition #CLASSIII CLASSIII|Data Rights Code|4|ENOVIAATRDEFAULT|| CLASSIII|Export Controlled|3|No|| #CLASSIV CLASSIV|Export Controlled|3|No|| CLASSIV|Data Rights Code|NOTDEFINED|ENOVIAATRDEFAULT|| #NoExportControl
MCADInteg-ClassificationType-Configuration	#Classification Types allowed for MCAD Classification. Syntax, Classification # Type|One or Many Classification Types|Mandatory|List of Classification objects Part Family|2|Y|CLASSIII~CLASSIV~NoExportControl General Class|1|Y|All

"Design TEAM Definition" policy needs to be modified to include a filter for having appropriate access privileges for applying classification information.

Block Save Enhancement

Collaboration Space information is required to check the access of the design.
To access the collaboration Space or Department information user need to perform some server side changes.

modify “DECSavePageDetails” table to add column project.

Using MQL script:

Run below MQL script to add “project” column in save dialog:

mod table DECSavePageDetails system column name project label project businessobject project setting 'Registered Suite' 'DesignerCentral'
setting 'Column Type' 'businessobject';

Using TCL script:

a. Open command prompt
b. Login to MQL
c. Run the below TCL file from the MQL command prompt.
d. >> run addColumnInDECSavePageDetails.tcl

addColumnInDECSavePageDetails.tcl

```
1  set context user creator;
2  trigger off;
3  tcl;
4  puts " "
5  puts "Business Object add column Script"
6  puts " "
7  mql mod table DECSavePageDetails system column name project label project businessobject project setting 'Registered Suite' 'DesignerCentral' setting
   'Column Type' 'businessobject';
8  puts " "
```

3DEXPERIENCE Connector for CreoParametric UserExits

Overview

User Exits are function calls that are implemented by the user and then called by the Connector for Creo Parametric to do specific operations before or after certain Connector for Creo Parametric functions.

The Connector for Creo Parametric provides user exits for these actions:

- ConnectPost
- CheckinDialogPre
- CheckinStart
- CheckinPartPre
- CheckinPartPost
- OpenPartPre
- OpenPartPost
- CheckoutPartPre
- CheckoutPartPost
- ModifyLockPartPre
- ModifyLockPartPost
- UpdateAttributesInCADPre
- UpdateAttributesInCADPost

Only those user exits that have been registered in the GCO will be called. User exits are registered by entering the name of the user exit into the IEF-UserExitEventRegistration attribute in the GCO. Multiple user exits are separated by a space.

For example, to register CheckinPartPost, CheckoutPartPre, and CheckoutPartPost you would set IEF-UserExitEventRegistration to “CheckinPartPre CheckinPartPost CheckoutPartPre CheckoutPartPost”.

The user exits may return an error code. A return code other than 0 indicates an error occurred in the user exit function. If an error is returned by the user exit function, the Connector for Creo Parametric’s operation is cancelled. (Note that this does not mean a transaction is rolled back.)

Possible uses for user exits are: image file or plot file generation; setting up attributes or parameters; or communicating with 3DEXPERIENCE via MQL commands. To assist with user exit communications with 3DEXPERIENCE, a function that can pass an MQL command through the X-CAD Design interface is exposed by the Connector for Creo Parametric.

The user exits are made available via a dynamically loaded shared library. Sample source code to a default library, a header file, and a sample makefile are provided. The name of the shared library (with its absolute path) must be defined using a new environment variable MXMCAD_USEREXITS_LIB.

Edit the Connector for Creo Parametric startup script in the bin directory to set the proper value for MXMCAD_USEREXITS_LIB. The default setting is:

```
MXMCAD_USEREXITS_LIB=
    %MXMCAD_HOME%\userexits\userExits\plat-win64\userExits.dll
```

Compiling & Linking User Exits

Follow the instructions below for setting up an environment for compiling and linking the user exit functions.

- Modify buildPe.bat to match your environment.
    a. set PROTOOL_SRC=<*pro toolkit directory*>
    b. set VS_DIR=<*to the base visual studio directory*>
    c. set SDK_DIR=<*to the base Microsoft SDK directory*>

- Run buildPe.bat to build the user exit shared library.
    a. Successful build creates userExits_pe.dll in /userexits/plat-win64 folder

To load the user exits at runtime set the environment variable MXMCAD_USEREXITS_LIB to the name of the shared library (with its full path). The recommendation is to do this in the startup script in <MXMCAD_HOME>\bin.

User Exit Functions

This section defines the function prototypes for each of the user exit functions. Note that all of the user exit function names begin with the prefix UE_.

Arguments are passed into the user exit functions with a ProArgument array. The ProE model, the Matrix id, type, name, and revision are passed for many of the functions.

In some cases, the user exit function may need to return an error message to be displayed by the Connector for Creo Parametric. This message will be displayed to the user upon completion of the operation. If the message string is not null, then the message string will be freed by the Connector for Creo Parametric. The user exit function must have allocated the string using malloc().

In addition, whenever a user exit function returns an integer value, a non-zero is interpreted as an error by the Connector for Creo Parametric.

UE_ConnectPost

UE_ConnectPost is called after the user logs into 3DEXPERIENCE and the Connector for Creo Parametric receives the contents of the GCO and the user's LCO. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError  UE_ConnectPost (ProArgument* inputs,
                                ProArgument** outputs)
```

UE_CheckinDialogPre

UE_CheckinDialogPre is called prior to displaying the checkin dialog. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_CheckinDialogPre (ProArgument* inputs,
                                  ProArgument** outputs)
```

A non-zero return will cancel the checkin operation.

UE_CheckinStart

UE_CheckinStart is called at the start of the checkin process after the user has selected Checkin from the checkin dialog. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_CheckinStart(ProArgument* inputs,
                                ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object (example: 'ProE Versioned Part')
inputs[4]: the Matrix revision of the object (example: 'A.0')

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
```

```
mxRev = inputs[4].value.v.w;
```

A non-zero return will cancel the checkin operation.

UE_CheckinPartPre

UE_CheckinPartPre is called after the business object has been created or updated in 3DEXPERIENCE but before the part file has been transferred. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_CheckinPartPre(ProArgument* inputs,
                                             ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object
inputs[4]: the Matrix revision of the object

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
```

A non-zero return will cancel the checkin operation.

UE_CheckinPartPost

UE_CheckinPartPost is called after the part file has been transferred to 3DEXPERIENCE. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_CheckinPartPost (ProArgument* inputs,
                                               ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object
inputs[4]: the Matrix revision of the object

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
```

```
mxRev = inputs[4].value.v.w;
```

UE_OpenPartPre

UE_OpenPartPre is called before the model is opened in the CAD system. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_OpenPartPre (ProArgument* inputs,
                                           ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the Matrix id of the object
inputs[1]: the Matrix name of the object
inputs[2]: the Matrix type of the object (example: 'ProE Part')
inputs[3]: the Matrix revision of the object (example: 'A.0')

They can be accessed as shown below.

```
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mxId = inputs[0].value.v.w;
mxName = inputs[1].value.v.w;
mxType = inputs[2].value.v.w;
mxRev = inputs[3].value.v.w;
```

UE_OpenPartPost

UE_OpenPartPost is called after the model is opened in the CAD system. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_OpenPartPost (ProArgument* inputs,
                                            ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object (example: 'ProE Part')
inputs[4]: the Matrix revision of the object (example: 'A.0')

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
```

UE_CheckoutPartPre

UE_CheckoutPartPre is called before a part file is checked out of 3DEXPERIENCE. Its function prototype is:

PRO_TK_DLL_EXPORT ProError UE_CheckoutPartPre(ProArgument* inputs,
ProArgument** outputs)

ProArgument* inputs contains the following:
inputs[0]: the Matrix id of the object
inputs[1]: the Matrix name of the object
inputs[2]: the Matrix type of the object
inputs[3]: the Matrix revision of the object

They can be accessed as shown below.

```
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mxId = inputs[0].value.v.w;
mxName = inputs[1].value.v.w;
mxType = inputs[2].value.v.w;
mxRev = inputs[3].value.v.w;
```

A non-zero return will cancel the checkout operation.

UE_CheckoutPartPost

UE_CheckoutPartPost is called after a part file is checked out of 3DEXPERIENCE. Its function prototype is:

PRO_TK_DLL_EXPORT ProError UE_CheckoutPartPost (ProArgument* inputs,
ProArgument** outputs)

ProArgument* inputs contains the following:
inputs[0]: the Matrix id of the object
inputs[1]: the Matrix name of the object
inputs[2]: the Matrix type of the object
inputs[3]: the Matrix revision of the object

They can be accessed as shown below.

```
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mxId = inputs[0].value.v.w;
mxName = inputs[1].value.v.w;
mxType = inputs[2].value.v.w;
mxRev = inputs[3].value.v.w;
```

UE_ModifyLockPartPre

UE_ModifyLockPartPre is called before an object is locked or unlocked. Its function prototype is:

PRO_TK_DLL_EXPORT ProError UE_ModifyLockPartPre(ProArgument* inputs,
ProArgument** outputs)

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object

inputs[3]: the Matrix type of the object (example: ‘ProE Part’)
inputs[4]: the Matrix revision of the object (example: ‘A’)
inputs[5]: an integer indicating lock or unlock

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
lock_operation = inputs[5].value.v.i;
```

The value of lock_operation will be 1 for lock and 0 for unlock.

UE_ModifyLockPartPost

UE_ModifyLockPartPost is called after an object is locked or unlocked. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_ModifyLockPartPost (ProArgument* inputs,
                                                  ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object (example: ‘ProE Part’)
inputs[4]: the Matrix revision of the object (example: ‘A’)
inputs[5]: an integer indicating lock or unlock

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
lock_operation = inputs[5].value.v.i;
```

The value of lock_operation will be 1 for lock and 0 for unlock.

UE_UpdateAttributesInCADPre

UE_UpdateAttributesInCADPre is called before attributes are updated in the CAD session. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_UpdateAttributesInCadPre(ProArgument* inputs,
                                                       ProArgument** outputs)
```

ProArgument* inputs contains the following:

inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object
inputs[4]: the Matrix revision of the object

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
```

A non-zero return will cancel the attribute update operation.

UE_UpdateAttributesInCADPost

UE_UpdateAttributesInCADPre is called after attributes are updated in the CAD session. Its function prototype is:

```
PRO_TK_DLL_EXPORT ProError UE_UpdateAttributesInCadPost (ProArgument* inputs,
                                                         ProArgument** outputs)
```

ProArgument* inputs contains the following:
inputs[0]: the ProE model
inputs[1]: the Matrix id of the object
inputs[2]: the Matrix name of the object
inputs[3]: the Matrix type of the object
inputs[4]: the Matrix revision of the object

They can be accessed as shown below.

```
ProMdl mdl;
wchar_t *mxId;
wchar_t *mxName;
wchar_t *mxType;
wchar_t *mxRev;
mdl = (ProMdl) inputs[0].value.v.p;
mxId = inputs[1].value.v.w;
mxName = inputs[2].value.v.w;
mxType = inputs[3].value.v.w;
mxRev = inputs[4].value.v.w;
```

External Functions

The Connector for Creo Parametric provides some utility functions that can be called by the user exits. These functions can be used to query for specific information from 3DEXPERIENCE and provide access to information within the Connector for Creo Parametric. These functions are defined in MxInterface.h.

MatrixCommand

This function sends an MQL command to 3DEXPERIENCE and returns the output. Any MQL command may be sent to 3DEXPERIENCE using this command. Its function prototype is:

int MatrixCommand(MXIF_APPLET *applet, char *mqlCommand, char **result);

Always pass NULL for the applet.

This function will return a non-zero value if 3DEXPERIENCE returns an error attempting to execute the mql command.

debugPrint

This macro prints a string in a specified format to the Connector for Creo Parametric.log file and to the screen. The binding for this function is similar to the C/C++ printf function. Its function prototype is:

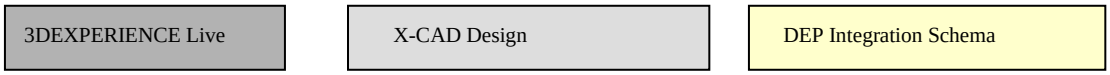
int debugPrint(char *msg, ...);

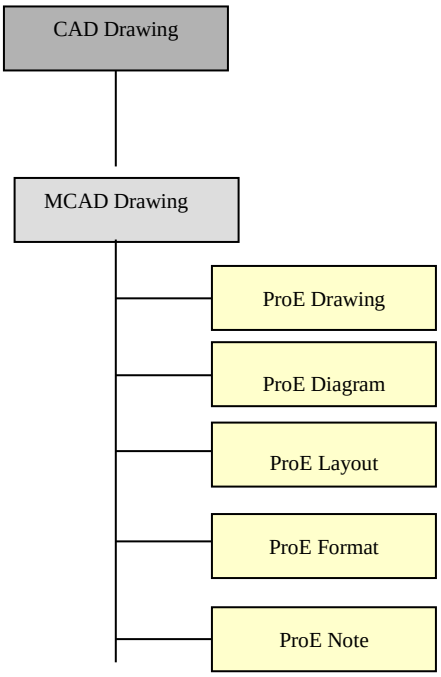
The msg string may include formatting controls as accepted by printf.

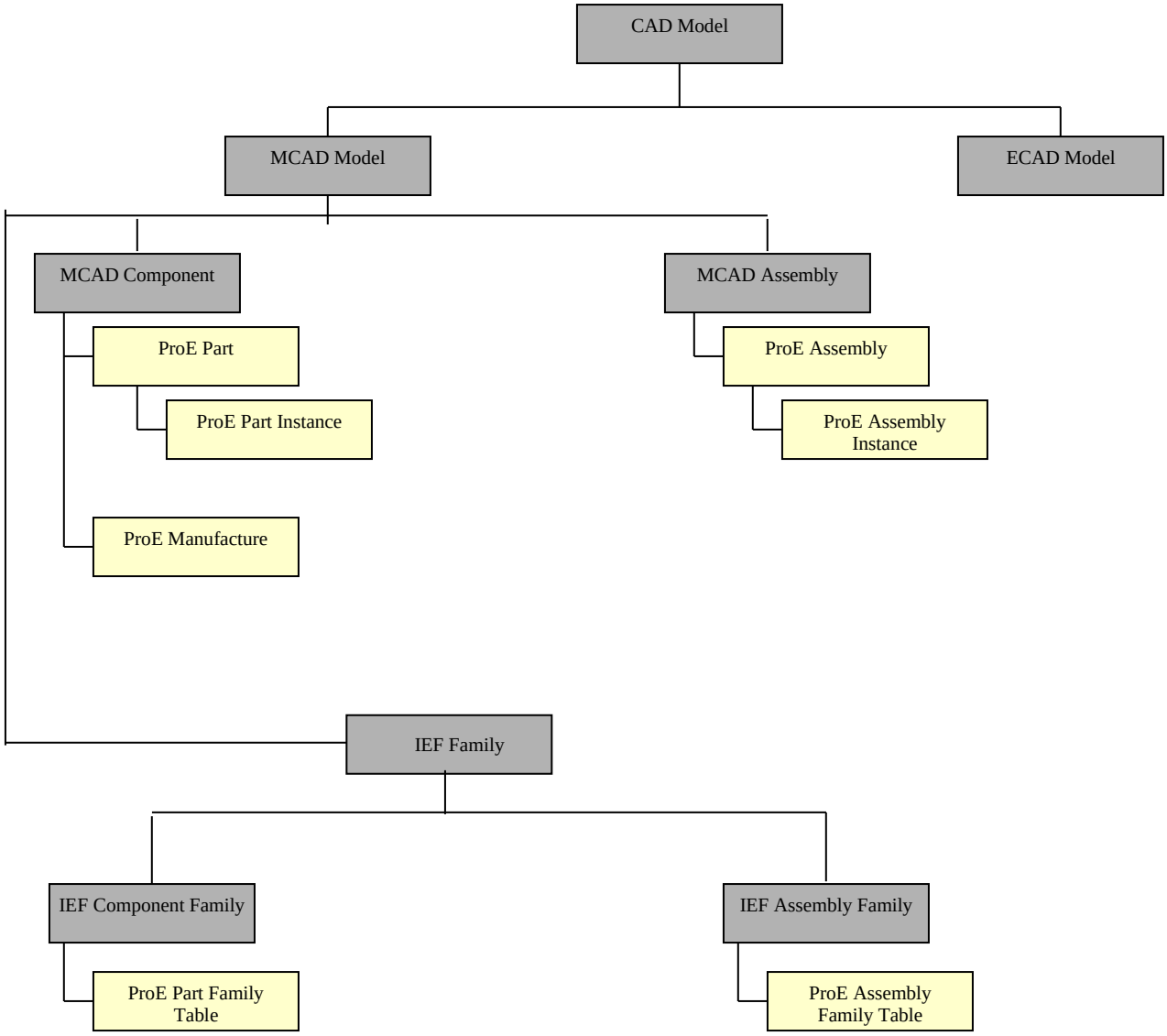
This example shows making an MQL call and printing the response using debugPrint.

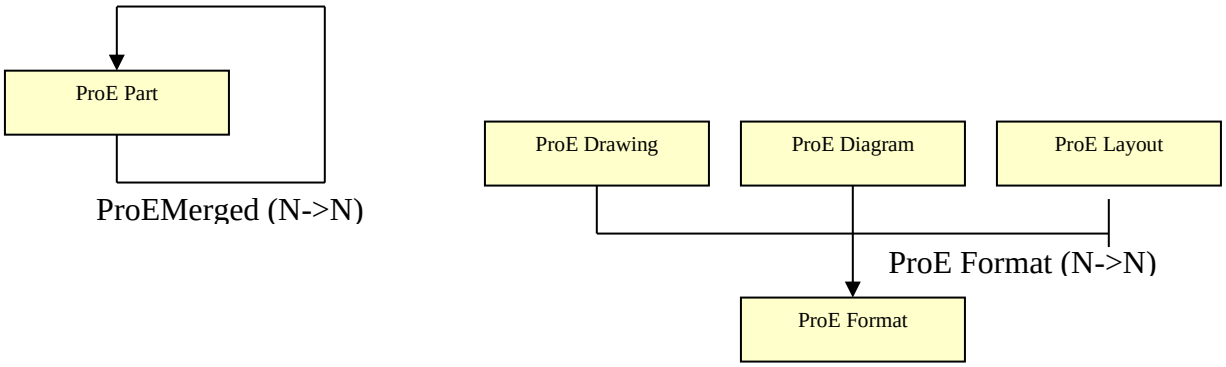
```
char *response = NULL;

MatrixCommand(NULL, "list person *", &response);
debugPrint("MQL command: list person *\n");
debugPrint("\tMQL response: %s\n", response);
```

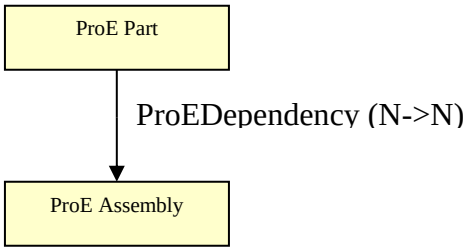
Appendix: Schema Diagrams

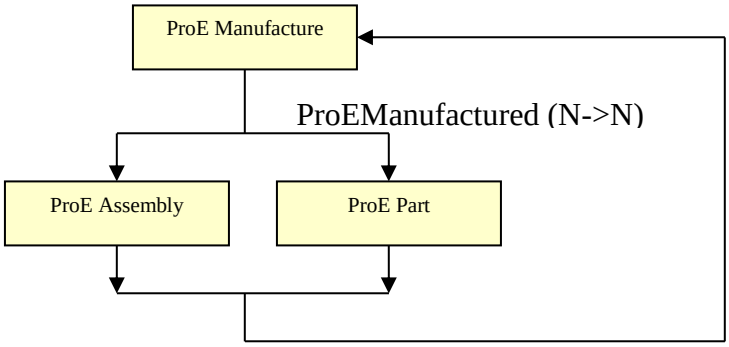


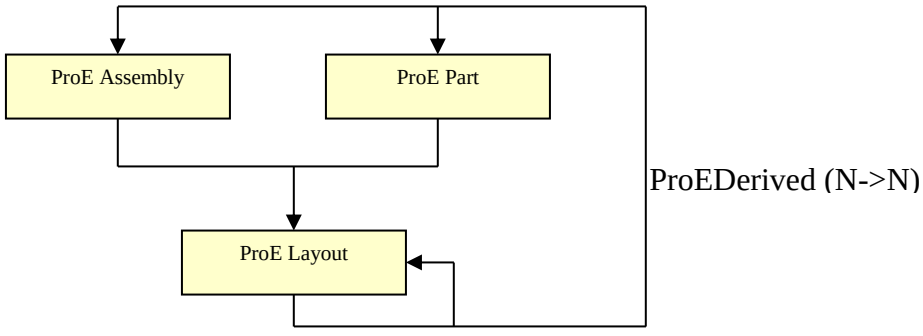


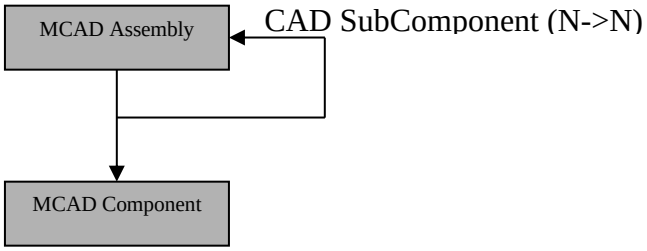

CAD Model
MCAD Model
ECAD Model
MCAD Component
MCAD Assembly
ProE Part
ProE Part Instance
ProE Assembly
ProE Assembly Instance
ProE Manufacture
IEF Family
IEF Component Family
IEF Assembly Family
ProE Part Family Table
ProE Assembly Family Table

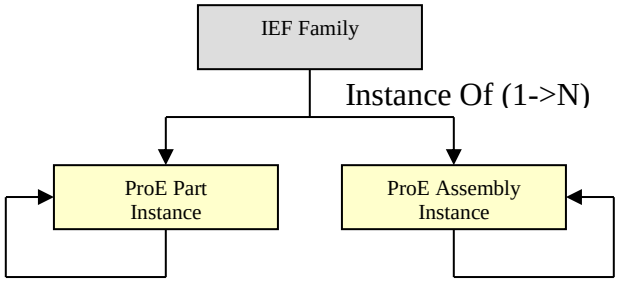

ProE Part
ProEMerged (N->N)
ProE Drawing
ProE Diagram
ProE Layout
ProE Format (N->N)
ProE Format

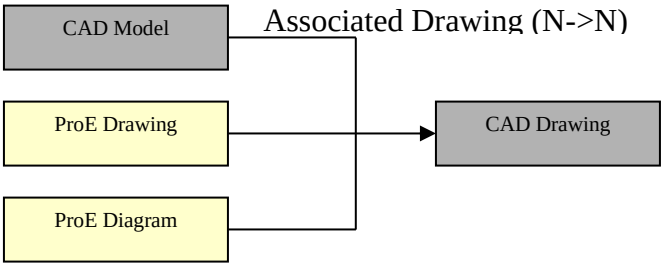

ProE Part
ProEDependency (N->N)
ProE Assembly


ProE Manufacture
ProEManufactured (N->N)
ProE Assembly
ProE Part


ProE Assembly
ProE Part
ProEDerived (N->N)
ProE Layout


MCAD Assembly
CAD SubComponent (N->N)
MCAD Component


IEF Family
Instance Of (1->N)
ProE Part Instance
ProE Assembly Instance


CAD Model
Associated Drawing (N->N)
ProE Drawing
CAD Drawing
ProE Diagram

Appendix: Deploying CREO Integration on a Network drive

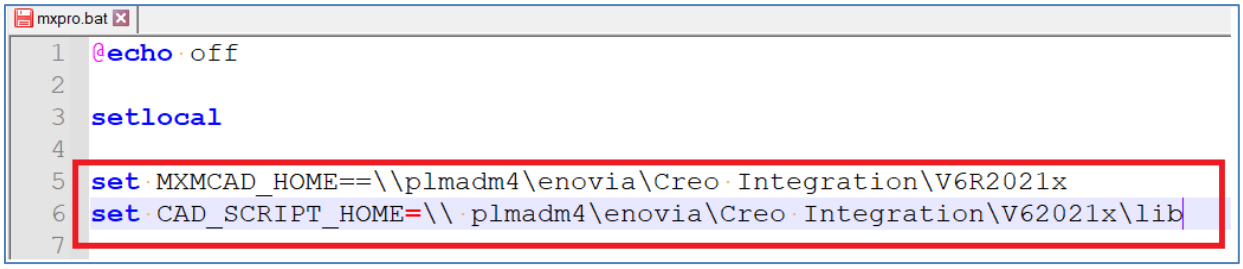
Pre-requisites:

- The below software must be installed prior to setting up the network integration installation
  - **Creo**
    Creo Parametric can be installed on the network or on the local machine.
  - **IEF Client**
    IEF Client must be installed on the local machine and cannot be installed to run over the network.
  - **Creo Integration**
    The Creo Integration first needs to be installed locally according to the ConnectorForCreoParamteric-EndUserInstallGuide. Once these steps are completed, the Integration can be copied over to the network location and configured.

1. Copy the Contents of the local Creo Integration Installation to the network location
   Example Location: \\plmadm4\enovia\Creo Integration\V62022x

2. Modify the following Creo Integration Network Installation files in a notepad:
   a. **Mxpro.bat** (located \\plmadm4\enovia\Creo Integration\V62022x\bin)

   set MXMCAD_HOME=\\plmadm4\enovia\Creo Integration\V6R2022x
   set CAD_SCRIPT_HOME=\\ plmadm4\enovia\Creo Integration\V62022x\lib



**If you are installing Creo over the network, verify that the following mxpro.bat settings are correct:

set path --- this setting includes the path to Creo. This will need to be modified if the Creo installation directory is now over the network.
set peBat="<path_to_Creo_Paramteric_Installation>\Parametric\bin\parametric.bat"

```
set CAD_SCRIPT_CFG=%CAD_SCRIPT_HOME%\cadscript.cfg
set CSI_PE_VER=creo6
set CAD_SCRIPT_EMBEDDED_MAIN_MODULE=%CAD_SCRIPT_HOME%\cmi\modules\EmbeddedMain.pyc
set PYTHONHOME=%CAD_SCRIPT_HOME%\python2.5\plat-win64
set PYTHONPATH=%MXMCAD_HOME%\modules\plat-win64;%CAD_SCRIPT_HOME%;%CAD_SCRIPT_HOME%\cmi\m
set path=%CAD_SCRIPT_HOME%\MxInterface\plat-win64;C:\cs\ProE\Creo 6.0.0.0\Parametric\bin;

rem set MXMCAD_STARTUP_ARGS=%*
rem Suppress command prompt window from appearing
set PTC_SUPPRESS_RESTART_AWARE=1

set peBat="C:\cs\ProE\Creo 6.0.0.0\Parametric\bin\parametric.bat"
```

b. **Cadscript.cfg** (located \\plmadm4\enovia\Creo Integration\V62022x\lib)

mxmcad=\\plmadm4\enovia\Creo Integration\V6R2022x\modules\Main

```
# alias for the program.  The path can use either UNIX or Win32 style
# directories.  The extension, if specified, is ignored and CADscript will
# try to load .py, .pyo, and .pyc variants of the file.

[embedded]
mxmcad=C:\ConnectorForCreoParametric\modules\Main

```

c. **Protk.dat** (located \\plmadm4\enovia\Creo Integration\V62022x\bin)

exec_file \\ plmadm4\enovia\Creo Integration\V6R2022x\lib\csi\pe\modules\plat-win64_csi_pe_creo6.dll
text_dir \\plmadm4\enovia\Creo Integration\V6R2022x\bin\text

**If there are other custom applications that are required (ex. eDrawing), add them below the Creo Integration entry after 'end'.

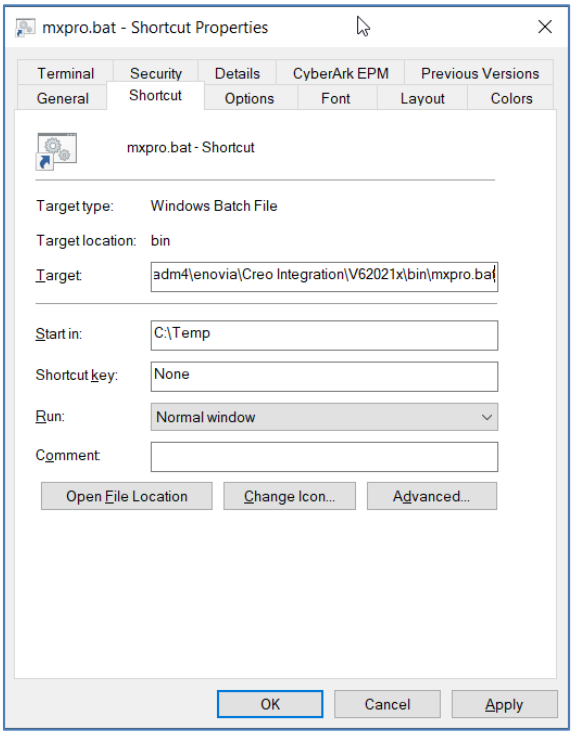
protk.dat

```
name _csi_pe_creo6
startup dll
exec_file \\plmadm4\enovia\Creo Integration\V62021x\lib\csi\pe\modules\plat-win64\_csi_pe_creo6.dll
text_dir \\plmadm4\enovia\Creo Integration\V62021x\bin\text
end
```

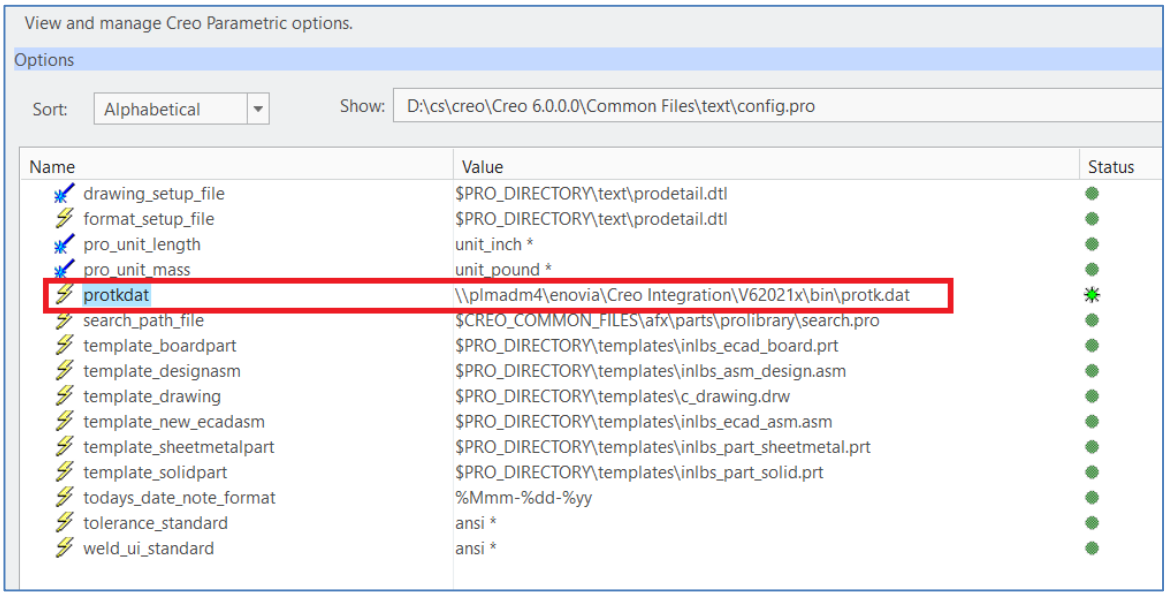
3. Create a desktop shortcut
   a. Copy the mxpro.bat file from \\plmadm4\enovia\Creo Integration\V62022x\bin
   b. Paste a shortcut to the desktop
   c. RMB on the shortcut > Properties
      - Target: "\\plmadm4\enovia\Creo Integration\V62022x\bin\mxpro.bat"

This must be in quotes if there are spaces in the path.

- Start in: Choose a location on your computer (Ex. C:\temp)



4. Config.pro
   a. The config.pro file will be located <path_to_Creo_Parameteric_Installation>\Common Files\text by default.
   b. Add the option 'protkdat' to the config.pro and set it equal to the network location of protk.dat (ex. \\plmadm4\enovia\Creo Integration\V6R2022x\bin\protk.dat)



Support

Support for the Connector for Creo Parametric is available through 3DEXPERIENCE by phone or email.

North America
Tel: +1 (877) 3D 4ALL1 (Toll Free)
+1 (877) 334 2551 (Toll Free)
+1 (704) 935 5690 (International)
Fax: +1 704 264 8888
E-mail: **AG.NA-Helpdesk@3ds.com**

EMEA
Austria +43 800 886 687
Belgium +32 2 357 53 03
France 0825 825 819
Germany 08001016391
Italy 02 334 306 306
Sweden 0771 320 710/+46 31 720 5858
Switzerland 0800 101 631
United Kingdom 0871 218 3467
Others countries +33 3 59 30 8419
Fax (EMEA): +33 1 70 73 4405
E-mail: Europe.SupportCenter@3ds.com
Languages: English, German, French

Asia Pacific
*Supported countries:*China PRC, Hong-Kong, Taiwan
Tel : (800)-810-3958
E-mail: PLMSupport.GCG@3ds.com
Languages: Chinese (Mandarin), English

Customer Support Center Korea
*Supported countries:*Korea
Tel :82-2-3270-7813
E-mail: PLMSupport.KOREA@3ds.com
Language: Korean, English

Customer Support Center India
*Supported countries***:** Australia, Indonesia, India, Malaysia, New Zealand, Philippines, Singapore, Thailand, Vietnam
E-mail: PLMSupport.INDIA@3ds.com
Language: English

Customer Support Center Japan
*Supported countries***:** Japan
Tel :+81 (0)3 5442 6051
E-mail: JP.Support.Tech@3ds.com
Language: Japanese, English

The integration creates a log file to help assess and diagnose problems or unexpected behavior: mxpro_debug.log. This file is located in the <user home directory>/mxpro_logs directory. It is very helpful to provide support with this file when the Connector for Creo Parametric produces unexpected results.

mxpro_debug.log contains Connector for Creo Parametric debugging statements. This file is only created if the debugPrintLog parameter is set to 1 in the settings.ini file. If the debugPrints parameter is set to 1 then trace statements will be written to the log file and if logSocketData is set to 1 then XML packets will be written to the log file.