PTC

PTC Windchill® Extension Guide

PTC Windchill® Extension 1.3

Contents

About This Guide

This guide documents the installation and use of the PTC Windchill Extension. It is intended for system administrators who are installing the PTC Windchill Extension and application developers who are using ThingWorx Composer to build a ThingWorx applications that connect to Windchill.

To take advantage of the functionality in the PTC Windchill Extension, you should have:

- Working knowledge of your Windchill instance
- Working knowledge of the ThingWorx Composer
- Ability to write Java script

Technical Support

Contact PTC Technical Support through the PTC website, or by phone, email, or fax if you encounter problems using this product or the product documentation.

The PTC Windchill eSupport portal provides the resources and tools to support your implementation:

https://support.ptc.com/appserver/cs/portal/

For complete support details, see the *PTC Customer Support Guide*:

http://support.ptc.com/appserver/support/csguide/csguide.jsp

You must have a Service Contract Number (SCN) before you can receive technical support. If you do not know your SCN, see "Preparing to contact TS" on the **Processes** tab of the *PTC Customer Support Guide* for information about how to locate it.

Comments

PTC welcomes your suggestions and comments on its documentation. To submit your feedback, you can send an email to documentation@ptc.com. To help us more quickly address your concern, include the name of the PTC product and its release number with your comments. If your comments are about this book, include the *PTC Windchill Extension Guide* book title.

1

Overview

This chapter describes the product functionality and requirements.

PTC Windchill Extension Product Overview

PTC Windchill Extension enables ThingWorx application developers to quickly and easily create connected applications that leverage the original product design information stored in Windchill. For example, you have the ability to do the following:

- Access Windchill BOM information.
- Access Windchill properties.
- Visualize product assemblies in 3-D.
- Get published formats associated with BOM items for connected products.
- Execute saved searched and queries.
- Execute Info*Engine tasks to acquire and create information in Windchill.

The PTC Windchill Extension includes:

- An extension bundle that is imported into ThingWorx Composer. For the list of extensions, see Installing Bundled Extensions on page 22.
- Sample mashup files that show the use of some entities contained in the extension bundle. For more information, see Using the Sample Mashups on page 41.
- A ThingWorx entities file that contains things, thing templates, shapes, and mashups that demonstrate the use of some entities contained in the extension bundle. These entities are intended as a sample only and should not be used to build your own mashup but rather serve as a template from which you can copy what you need. For more information, see Using the Sample Mashups on page 41.
- A set of Windchill patches that correspond to specific Windchill releases. The code in each patch provides the hooks that allow a ThingWorx application to connect to the corresponding Windchill release. For more information, see Installing PTC Windchill Patches on page 25.

For detailed information on all included files, see PTC Windchill Extension Media on page 16.

Product Requirements

To use the PTC Windchill Extension, you must be using the following products:

- PTC ThingWorx 6.5
- Windchill at one of the following release levels:
    - 10.2 M020 CPS 06 or later
    - 10.2 M030 CPS 01 or later

○ 11.0

To connect to systems running the Windchill releases, you must install the Windchill patch that corresponds with the release and is included with PTC Windchill Extension.

• Security fixes for PTC HTTP Server (bundled with Windchill) are required when using SSL for two-way secure communication between a ThingWorx application and Windchill. For more information, see Recommended Configurations on page 28. The required Windchill security fixes are available from PTC:

○ For Windchill 10.2 M020, update to CPS06 or later for required security fixes.

○ For Windchill 10.2 M030 and later releases, the PTC HTTP Server bundled with the release has the required security fixes for using SSL.

PTC recommends that you install the CPS or Early Release Download with the option to **overwrite** the PTC HTTP Server configuration and redo the configuration. If you choose to install the PTC HTTP Server with the option to **preserve** the configuration (the default option), then you need to merge the configuration scripts before you can configure SSL. For more information, see Example Configuration Using SSL for Secure Communications on page 71.

• PTC Creo View 3.0 M031 is required to use the Creo View widget in a mashup.

Entity Overview

This section offers a high level overview of important entities provided with the files included in the PTC Windchill Extension.

`ptc-windchill-extension`

Entity	Description
WindchillConnector	Template representing an interface to a Windchill system. Includes services that provide the means to access Windchill data and the processing power of the Windchill system.

`ptc-sample-mashup`

Entity	Description
ptc-part-demo-shape	Shape of the data shown in the tree and the properties panel of the ptc-connected-demo-mashup and the search results of the ptc-search-demo-mashup.
ptc-windchill-demo-	Shows how to wrap calls to Windchill services with

Entity	Description
thing	ThingWorx services that are used from a mashup. The services defined in this Thing are used by the sample mashups.
ptc-connected-demo-mashup	A sample mashup that demonstrates the retrieval of a Windchill product structure based on the input of a Windchill part, Windchill document, or EPM document number. The mashup also emulates the inclusion of data coming from an ERP system.
ptc-search-demo-mashup	A sample mashup that demonstrates the use of Windchill search capabilities.

For more information, see .

dataquery

Entity	Description
com.ptc.dataquery. DataQueryTemplate	Template to create data queries for connected PTC systems
com.ptc.dataquery. PTCConnectorTem-plate	Template to create Things for connecting to any PTC systems
ptc-demo-navigation	Generated Thing that provides the service to navigate a part structure
ptc-navigation-demo-shape	Generated DataShape that defines the shape of the results of part structure navigation
WindchillConnector-NavParams	Defines the set of parameters that Windchill navigation requires

For more information, see .

Additional Entities

After installation and configuration is complete, you can browse the ThingWorx Composer to learn more about the entities included with the PTC Windchill Extension. For example, to learn more about the WindchillConnector Template, open it in the Composer and view its list of services. As shown in the screen below, you can hover over a service to view additional information about the service.

Architecture Overview

The following diagram illustrates that a portion of the PTC Windchill Extension runs inside ThingWorx and a portion runs inside Windchill. These pieces work together to provide secure access to Windchill data and services.

After you have imported PTC Windchill Extension into the ThingWorx Composer, a set of extensions is ready to use. The extensions allow you to create mashups that display data from Windchill. Using the extension, you can use widgets to create mashups and applications that display data from Windchill. For more information, see Using the Sample Mashups on page 41.

Connecting to a Windchill Instance

To connect a ThingWorx application to a Windchill instance, Windchill must have specific code that allows the connection.

The following diagram shows the secure interaction between the PTC Windchill Extension code in ThingWorx and the PTC Windchill Extension code in Windchill.

The required versions of ThingWorx, Windchill, and other related software is discussed next in Product Requirements on page 8.

2

Installing PTC Windchill Extension

This chapter contains download and media details as well as installation information.

Downloading the PTC Windchill Extension

The PTC Windchill Extension is available from the ThingWorx Marketplace:

 http://marketplace.thingworx.com

From the welcome page, click **SIGN IN**.

📝 Note

Your company must have purchased ThingWorx and you must have a valid ThingWorx user name and password to download this extension. If you do not have a valid ThingWorx user name and password, you can have your system administrator request one for you by sending the request to `twc@thingworx.com`.

After you have signed in, the page refreshes and all extension categories display. Locate and download the PTC Windchill Extension. For example, navigate to the **ENTERPRISE SYSTEM** category and click the extension:
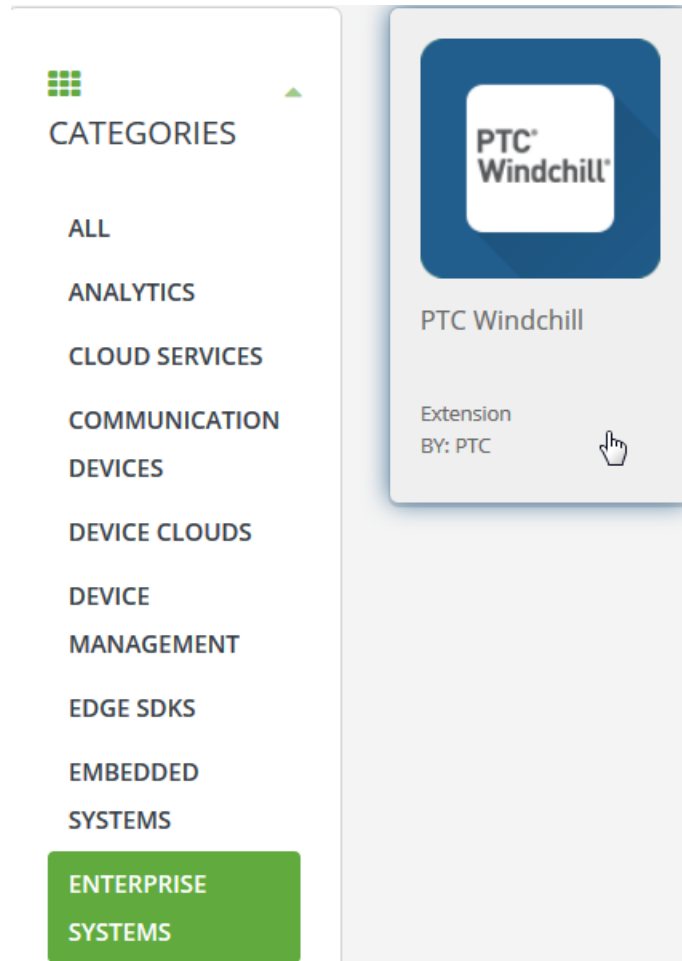
From the downloads page, you can access a description of the extension by clicking the **Details** tab and download the PTC Windchill Extension ZIP file by clicking **DOWNLOAD**.

PTC Windchill Extension Media

The PTC Windchill Extension ZIP contains the following files:

- Windchill patches for each supported release that enables ThingWorx applications to connect to Windchill. For more information, see Installing PTC Windchill Patches on page 25.
- Extensions in `ptc-windchill-extension-{version}.zip`. For more information, see Installing Bundled Extensions on page 22.

  - `ptc-windchill-connection-extension.zip`

    PTC Windchill Connection Extension enables connectivity with Windchill and includes the WindchillConnector Thing Template.

  - `core-ui-ext.zip`

    PTC Core UI Library that includes additional widgets.

  - `ptc-creo-view-extension.zip`

    PTC Creo View Extension is the Visualization widget extension. For more information, see Using PTC Creo View Extension on page 59.

  - `infotableselector_ExtensionPackage.zip`

    InfoTable Selector Extension includes the InfoTable Selector Widget which is used to manage selected rows of an InfoTable.

  - `ptc-identity-provider-authenticator-extension.zip`

    PTC Identity Provider Authenticator Extension is used for single sign-on with Windchill. For more information, see Architecture of Single Sign-On on page 75.

  - `dataquery-ext.zip`

    PTC Data Query Extension provides a quick and easy way to access PTC data through ThingWorx. The PTC Data Query Extension has some support limitations. For information on limitations and other information about the extension, see Using PTC Data Query Extension when Developing Your Application on page 79.

  - `UniversalConverter_Extensions.zip`

    Universal Converter, which is also available through the ThingWorx Marketplace.

  - `twx-connector-ext.zip`

    PTC AngularJS ThingWorx Connector Extension

- `ptc-identity-provider-authentication-filter-{version}.jar`

The PTC Identity Provider Authentication Filter is a servlet filter that is used for single sign-on with Windchill. For more information, see Architecture of Single Sign-On on page 75.

- `ptc-sample-mashup-{version}.xml`

  A ThingWorx Entities file that contains Templates, Things, Shapes, and Mashups that demonstrate use of the entities contained in the extension bundle. These entities are intended as a sample only and should not be used to build your own mashup but rather serve as a template from which you can copy what you need. For information about using the sample mashups, see Using the Sample Mashups on page 41.

- `README.txt`

  Contains brief information regarding the PTC Windchill Extension with a pointer to this guide.

Installing the PTC Windchill Extension in ThingWorx Composer

The following sections describe the actions needed to complete the installation of PTC Windchill in the ThingWorx Composer. All actions described in these sections require that you are logged in as an administrator.

1. Deleting Existing Entities and Extensions Before Installing PTC Windchill Extension on page 17
2. Installing Bundled Extensions on page 22
3. Installing Sample Mashups on page 24
4. Installing PTC Windchill Patches on page 25

Deleting Existing Entities and Extensions Before Installing PTC Windchill Extension

If you have previously installed the PTC Windchill Extension, continue reading this section to learn how to find and delete existing entities and extensions.

If you have not previously installed the PTC Windchill Extension, you can skip this section.

⌨ Note

Failing to remove the entities and extensions before importing updates can cause problems with the updated files. This is because data can be stored in Tomcat memory that is associated with the old entity and this data is not removed unless you delete the entities and restart Tomcat.

Because there are dependencies among the entities and the extensions bundled with the PTC Windchill Extension, the order in which you delete existing entities and extensions is important. Complete the following steps:

1. Delete all the demo entities.

   For more information, see Searching for Demo Entities to Delete on page 19.

2. Delete the **ptc-windchill-connection-extension** extension.

   For deletion instructions, see Deleting Extensions on page 21.

3. Delete the **dataquery-ext** extension.

4. Optionally delete the other extensions:

   core-ui-ext

   ptc-creo-view-extension

   ptc-identity-provider-authenticator-extension

   twx-connector-ext

5. If you have an existing extension called **ptc** that contains the initial Creo View widget, delete that extension. Leaving the old Creo View extension causes problems with the new **twx-connector-ext** extension.

6. Restart Tomcat to clear the Tomcat memory

Searching for Demo Entities to Delete

To locate the demo entities that need to be deleted from the ThingWorx Composer, display all entities and then use the **Advanced** search button. Populate the **Tags** field with **Applications ptc-connedcted-plm-demo** using the magic wand icon. For example:
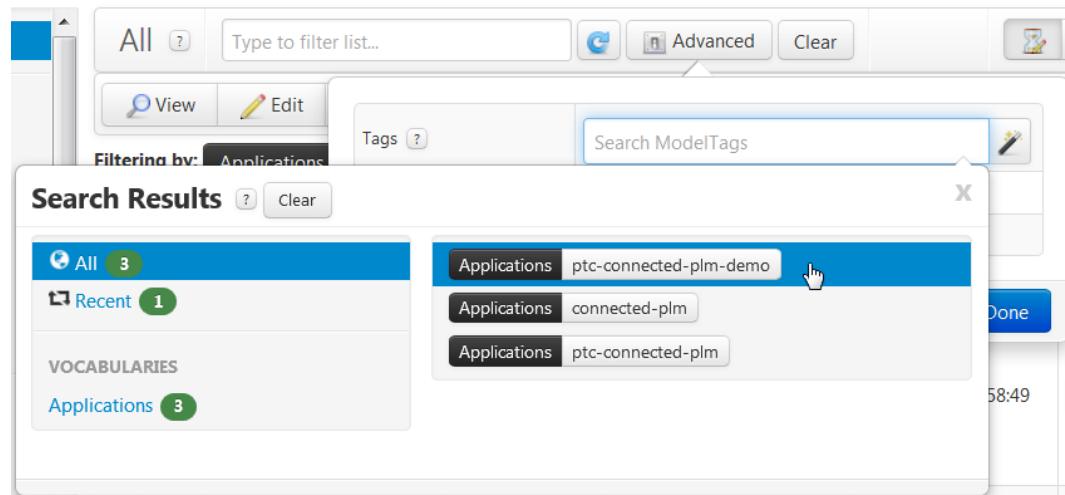
The resulting list of tagged entities is displayed. Select all of the entities and click **Delete**.

	View	Name	Description	Type
☑	🔍	ptc-connected-demo-...	ptc-connected-demo-mashup	☐ Mashup
☑	🔍	ptc-windchill-demo-thi...	Shows how services can be wrapped to have a custom dataShape used in the mashups.	Thing
☑	🔍	ptc-demo-navigation	Generated Thing that provides the service to navigate a part structure.	Thing
☑	🔍	ptc-demo-low-quantity	ptc-demo-low-quantity	StateDefinition
☑	🔍	ptc-configuration-dem...	ptc-configuration-demo-shape	DataShape
☑	🔍	ptc-part-demo-shape	ptc-part-demo-shape	DataShape
☑	🔍	ptc-demo.wt.part.WTPart	ptc-demo.wt.part.WTPart	DataShape
☑	🔍	ptc-demo-cost-data	Demo server that returns hard-code cost data.	DataShape
☑	🔍	ptc-navigation-demo-s...	Generated DataShape that defines the shape of the results of part structure navigation.	DataShape
☑	🔍	ptc-demo.wt.change2.R...	ptc-demo.wt.change2.ReportedAgainst	DataShape
☑	🔍	ptc-demo.wt.change2....	ptc-demo.wt.change2.WTChangeIssue	DataShape

Filtering by: Applications ptc-connected-plm-demo **Exclude** *System Objects*

Deleting Extensions

To delete the extensions, complete the following steps:

1. From the **Import/Export** menu in the ThingWorx Composer, select **Extensions** ▶ **Manage**. For example, the following screen shows the actual names and descriptions of the extensions that have been installed:

Installed Extension Packages	✖ Delete Extension
Name	**Description**
UniversalConverter_Extensions	UniversalConverter Extensions
ptc-windchill-connection-extension	Extension to enable connectivity with Windchill
infotableselector_ExtensionPackage	infotableselector Widget
core-ui-ext	PTC Core UI Library
ptc-identity-provider-authorization-extension	PTC Identity Provider Authorization Extension
twx-connector-ext	PTC AngularJS Thingworx Connector
ptc-creo-view-extension	PTC Creo View Widget
dataquery-ext	PTC Data Query

2. Select an extension name and then click **Delete Extension**.

Installing Bundled Extensions

To install the extensions included with the PTC Windchill Extension, complete the following steps:

1. From the **Import/Export** menu in ThingWorx Composer, select **EXTENSIONS ▸ Import**:

2. In the **Import Extensions** window, click **Browse** and navigate to `ptc-windchill-extension-{version}.zip`.

3. Click **Import**.

4. If prompted, refresh the composer.

To view the extensions after the import is complete, select **Import/Export ▶ Manage**. The following screen shows the names and descriptions of the extensions that have been installed:

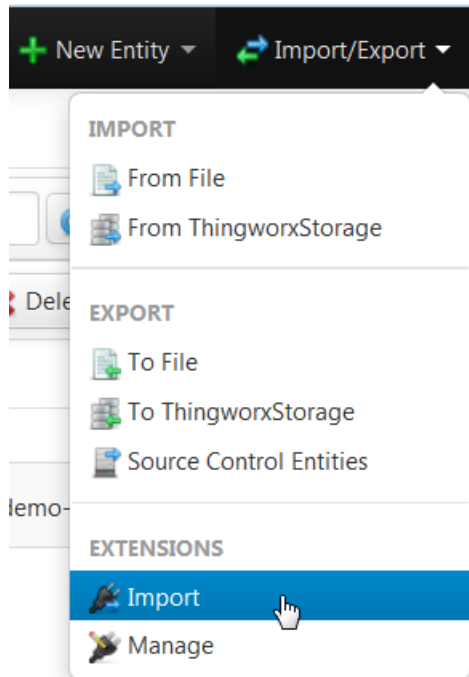| Name	Description
UniversalConverter_Extensions	UniversalConverter Extensions
ptc-windchill-connection-extension	Extension to enable connectivity with Windchill
infotableselector_ExtensionPackage	infotableselector Widget
core-ui-ext	PTC Core UI Library
ptc-identity-provider-authorization-extension	PTC Identity Provider Authorization Extension
twx-connector-ext	PTC AngularJS Thingworx Connector
ptc-creo-view-extension	PTC Creo View Widget
dataquery-ext	PTC Data Query

Selecting a row in the **Installed Extension Packages** table displays the entities in the extension details under **Package Details**. For example, selecting **ptc-windchill-connection-extension** displays the following details:

Package Details

Name	Description
WindchillConnector	Template representing a Windchill Instance interface.
WindchillConnectorThing	
IEPropertyDescription	Describes attributes returned by Info*Engine DescribeTypes service.
IESchemaNode	Describes attributes returned by Info*Engine ListTypes service.
IEListBoolean	Use when Info*Engine returns a Boolean list.
ptc-creoview-download-links	ptc-creoview-download-links
IEContentHandle	Use when Info*Engine returns a content handle.

Installing Sample Mashups

To install the sample mashups included with this release, complete the following steps:

1. From the **Import/Export** menu in the ThingWorx Composer, select **IMPORT ▸ From File**:
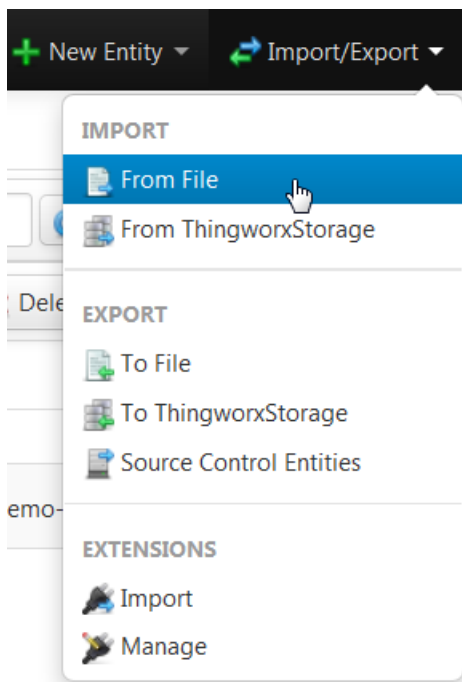
2. In the **Import From File** window, select **Entities** and click **Browse**.
3. Navigate to `ptc-sample-mashup-1.0.0-13.xml` and click **Import**.

The sample mashups appears under **Mashups**:
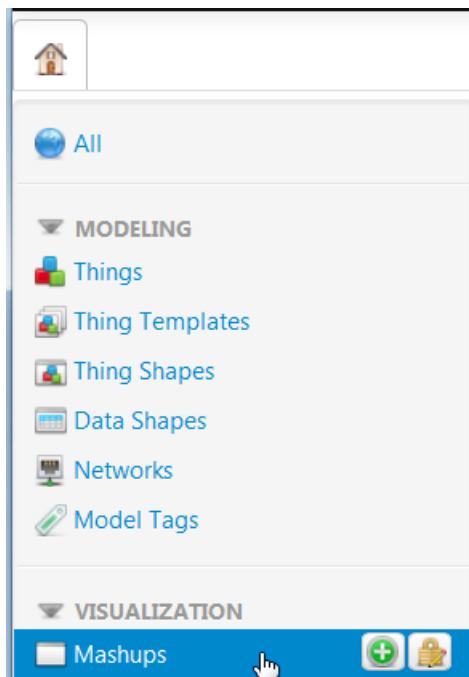
To configure the sample mashup, you must enter the base URL in the **ptc-windchill-demo-thing** configuration. For configuration details, see .

Installing PTC Windchill Patches

The required PTC Windchill patch adds several web service endpoints that allow ThingWorx to interact with Windchill.

The name of the ZIP file identifies the release in which it should be installed. For example, on a 10.2 M020 system, install the patch in `10.2-M020_ WCTWXExtension_{version}.zip`.

Installing the Patch

Each patch includes instructions specific to the release. To install the patch, unzip the ZIP file and use the instructions found in the `Readme` file that is included with the patch.

10.1 M040 Special Instructions

If you are on Windchill 10.1 M040, you may need to modify the `wt.epm.displayCalculatedLink` property in `wt.properties` after the patch is installed.

No changes are required if the property is not set or is set to `true`; however, if the property is set to `false`, the **View Drawing** app will not function unless the property is changed to `true`. Consider whether changing this property setting has additional impacts at your site.

If your site chooses to set the preference to `true`, use the following procedure to update `site.xconf` after the patch is installed.

1. Enter the following lines in `site.xconf`:
   ```
   <Property name="wt.epm.displayCalculatedLink" overridable="true"
        targetFile="codebase/wt.properties" value="true"/>
   ```

📝 **Note**

Code examples in this guide have been reformatted to fit the page and may contain line numbers, hidden editing characters (such as tabs and end-of-line characters) and extraneous spaces. If you cut and paste code, check for these characters and remove them before attempting to use the example in your application.

---

2. Run `xconfmanager –p` to populated the properties.

3

Configuring Your System

This chapter contains configuration considerations for your system.

All actions described in this chapter require that you are logged in as an administrator.

Recommended Configurations

Not all configuration steps are necessary in every installation of the PTC Windchill Extension. As a way to simplify the installation and configuration, PTC recommends configurations for the following environments that only configure what is most reasonable:

* Production
* Development
* Demo

Each recommended configuration contains links to the sections that have the detail steps to follow. Choose a recommended configuration and follow the links to the steps and perform the configuration.

Production Configuration

The recommended production environment has the following characteristics:

* Encrypted connection between Windchill and ThingWorx. Encrypted communications is configured in the next step as part of using SSL to establish a trusted relationship.
* Trust between Windchill and ThingWorx that is based on SSL public/private keys. Follow the steps in Using SSL for Secure Communication on page 29.
* A single sign-on prompt for mashup users. The single sign-on uses Windchill credentials. Follow the steps in Configuring ThingWorx Single Sign-On when Connecting to Windchill on page 33.

Development Configuration

The recommended development environment has the following characteristics:

* Non-encrypted communication between Windchill and ThingWorx.
* Trust between Windchill and ThingWorx that uses the Trusted Host mechanism. Follow the steps in Using Trusted Host on page 32.
* Single sign-on is not required.

  Mashup users will be prompted separately for ThingWorx credentials and Windchill credentials. Not using single sign-on will only work if the authenticated ThingWorx user also exists as a Windchill user.

Demo Configuration

The recommended demo environment has the following characteristics:

* Non-encrypted communication between Windchill and ThingWorx.
* Trust between Windchill and ThingWorx that uses the Trusted Host mechanism. Follow the steps in Using Trusted Host on page 32.

- A single sign-on prompt for mashup users. The single sign-on uses Windchill credentials. Follow the steps in .
- Sample mashups. Fore more information, see .

Using SSL for Secure Communication

Your site can use two-way SSL to secure the communication between the ThingWorx server and the Windchill application. As described in , PTC recommends using SSL when working in a production environment. For development and demo environments, see . The extension can use SSL to both mutually authenticate the servers to each other and protect the communication itself.

An SSL connection requires that both systems trust each other; to do this the **WindchillConnector** thing template must be configured to reference Java key and trust stores held on the ThingWorx server which provide keys for the transaction. The HTTP Server on the Windchill server must be configured to trust those keys.

SSL configurations vary considerably and this guide does not attempt to describe all options available in an SSL configuration. Instead, the following steps give an overview of the process for configuring SSL, and a detailed example is available in .

This procedure assumes that both ThingWorx and Windchill are configured to use SSL for their standard communications. It also relies on configuration scripts that come with the most recent version of the PTC HTTP Server. The most recent PTC HTTP Server is included with Windchill 11.0 and the most recent CPS releases for Windchill 10.2.

> **Note**
>
> It is not necessary to use the same SSL key for the primary Windchill or ThingWorx communication and for the ThingWorx to Windchill communication discussed here. For example, a commercial trusted wildcard certificate could be used for Windchill and a self-signed certificate used between ThingWorx and Windchill.

Configuring ThingWorx

ThingWorx requires Java key stores and trust stores.

- Create them on the same server as ThingWorx.
- Create them on the local file system.
- Reference them in the configuration of the **WindchillConnector** thing template used to connect ThingWorx to the Windchill system being secured.

SSL Connection Configuration

Name	Value
SSLKeyStoreType	JKS
SSLKeyStorePath	/usr/local/tomcat/twx.jks
SSLTrustStorePath	/usr/local/tomcat/trustedstore_ptc.jks
SSLKeyStorePassword	Change Password
SSLTrustStorePassword	Change Password
SSLTrustStoreType	JKS

1. For each of the stores, specify the type and the password used to access the store.

> 📋 **Note**
>
> It is recommended but not mandated that the type is JKS.

2. Into the key store, import the private key used by Windchill.
3. Into the trust store, import the certificates used to establish trust credentials for those keys.

Configuring Windchill

On the Windchill server, configure SSL authentication in both the PTC HTTP Server and the PTC Embedded Servlet Engine to trust the ThingWorx key and certificate.

> **📝 Note**
>
> Code examples have been reformatted to fit the page and may contain line numbers, hidden editing characters (such as tabs and end-of-line characters) and extraneous spaces. If you cut and paste code, check for these characters and remove them before attempting to use the example in your application.

1. Update PTC HTTP Server configuration to reference the CA certificates file.
   - For Windchill 10.2 releases:
      a. Navigate to and edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.
      b. Uncomment the `SSLCACertificateFile` entry so that it refers to the `ca-bundle.crt` file. Make a note of the location of the `ca-bundle.crt` file.
      c. Save changes.
   - For Windchill 11.0 releases:
      a. Create a file in `<APACHE_HOME>\conf\ca-bundle.crt`. This location is recommended but not required.
      b. Paste the contents of `Server.crt` file used in HTTP Server.
      c. Save `ca-bundle.crt`.
      d. Create a `sslclientauth.conf` file at `<APACHE_HOME>\conf\sslvhostconf.d`.
      e. Add SSLCACertificateFile <PATH_TO>\ca-bundle.crt in it so that it refers to the ca- bundle.crt file.
      f. Save changes.
2. Add the ThingWorx certificate to PTC HTTP Server's list of trusted CA certificates.
   a. Edit the `ca-bundle.crt` file you pointed to previously in the `SSLCACertificateFile` entry.

   > **📝 Note**
   >
   > If the file does not yet exist, create the file and any required directories.

   b. Append the contents of the ThingWorx certificate held in the trust store to this file.
   c. Save changes.
3. Configure PTC HTTP Server for the sslClientAuth URL.

a. Find the Windchill web app name in `<windchill-home>/codebase/wt.properties` in the entry `wt.webapp.name`.

b. Open a shell or command prompt and enter the following:

```
cd <HTTPSERVER_HOME>ant -f webAppConfig.xml -DappName=[windchill-web-app]
     -Dresource=sslClientAuth -DresourceAuthType=sslClientAuth addAuthResource
```

c. Configure Windchill to trust the ThingWorx certificate.

   i. Edit `<windchill-home>/codebase/WEB-INF/web.xml`.

   ii. Find `<filter-name>TrustedSSLAuthFilter</filtername>`.

   iii. Add an additional *<init-param>* element after the existing one where *[thingworx-cert-name]* is the name of the ThingWorx certificate:

```
<init-param><param-name>trustedSubjectPattern.1</param-name><param-value>
     [thingworx-cert-name]</param-value></init-param>
```

d. Save changes.

Using Trusted Host

When creating an application for internal use (such as development, testing, or demos), you can configure communication using just a trusted host mechanism. For production environments, see Using SSL for Secure Communication on page 29.

Trusted hosts are established through the `wt.auth.trustedHosts` property in the `wt.properties` file on your Windchill system. You must ensure that the host name of the ThingWorx server has been added to this property.

If the `wt.auth.trustedHosts` property exists, you can use the `-d` parameter on the `xconfmanager` command to display the existing values of the property. For example, enter a command similar to the following from a Windchill shell:

```
xconfmanager -d wt.auth.trustedHosts=twxhost1 -t codebase/wt.properties
```

If needed, you can enter an xconfmanager command from a Windchill shell to add the ThingWorx host name:

• If the property is not set in `wt.properties` and the host name is `twxhost1`, enter a command similar to the following to set the property to the host name:

```
xconfmanager --set wt.auth.trustedHosts=twxhost1 -t codebase/wt.properties -p
```

• If the property has already been set in `wt.properties` and the host name is `twxhost1`, enter a command similar to the following to add the host name:

```
xconfmanager --add wt.auth.trustedHosts=twxhost1 -p
```

• When multiple hosts exist, enter as a single value similar to the following example:

```
xconfmanager -s wt.auth.trustedHosts="hostname1 hostname2 hostname3"
```

For more information, see the related PTC case solution. https://support.ptc.com/appserver/cs/view/solution.jsp?n=CS182327

Configuring ThingWorx Single Sign-On when Connecting to Windchill

ThingWorx users must have corresponding user identities in Windchill to access mashups that use the services provided by the extension. PTC recommends using the single sign-on option to address this.

The single sign-on solution uses Windchill as the identity provider for ThingWorx. A user that opens a browser to a mashup or any ThingWorx URL is routed to Windchill for authentication. Once authenticated, the browser is routed back to the mashup or another page in ThingWorx and the user is able to access ThingWorx or the mashup as the user authenticated in Windchill. For technical details, see Architecture of Single Sign-On on page 75.

Single sign-on requires that the same users exist in both Windchill and ThingWorx. PTC Navigate provides an option to automatically create users in ThingWorx once they have been authenticated in Windchill. If this option is not enabled or single sign-on is not used, then users must independently exist in both Windchill and ThingWorx.

Use the steps in the following sections to configure single sign-on using Windchill as the identity provider.

Configuring the PTC Identity Provider Authenticator

The following steps describe how to configure the PTC Identity Provider Authenticator.

1. Select **Authenticators** under **SECURITY** in the left navigation pane.
2. Click the **ptc-identity-provider-authenticator** link to display the general information about the extension.
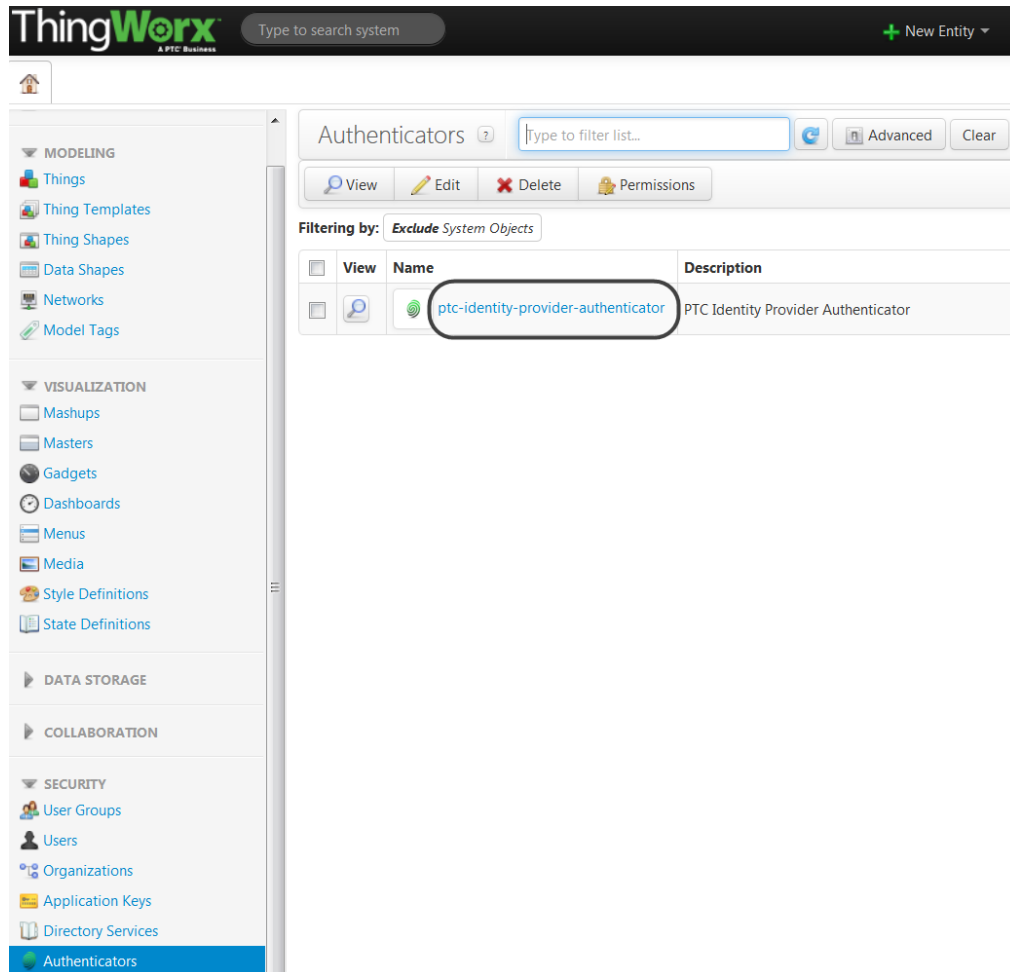

The **General Information** page opens.

3. On the **General Information** page, complete the following steps:
   a. Select the **Enabled** checkbox.
   b. Enter a value in the **Priority** field. By default, the priority is 1 indicating that this authenticator is the first authenticator to run.

      The value set in the **Priority** field is important if you want to implement checks by multiple authenticators. If the authenticator with a priority of 1 fails, then the next authenticator does the authentication check and so on.

4. Click **Configuration** under **ENTITY INFORMATION** in the left navigation pane:
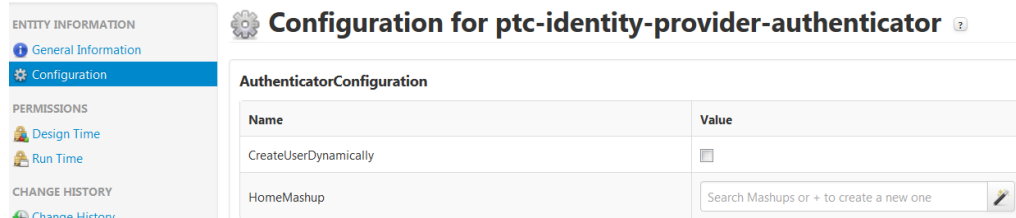
Use the two options, **CreateUserDynamically** and **HomeMashup**, with single sign-on so that users who are authenticated in Windchill are automatically added as users to ThingWorx and assigned a home mashup. When a user opens a browser to ThingWorx for the first time, they will be routed to Windchill for authentication. Once authenticated, a corresponding user is created in ThingWorx and the user is assigned a home mashup. Then the browser is routed to the specified home mashup.

• **CreateUserDynamically** – When selected, configures the authenticator to automatically create a user in ThingWorx if the user does not yet exist and the user has been authenticated by Windchill. If the authenticator is not configured to automatically create users, the browser is still routed to Windchill for authentication but then fails to open ThingWorx if the users does not exist.

• **HomeMashup** – Assigns a home mashup to the newly created users. If no home mashup is specified, ThingWorx routes the browser to the ThingWorx search page.

For PTC Navigate, set to **LandingPageAccessAppMashup**.

5. Ensure that a home mashup value is set for either all dynamically created users using the **HomeMashup** field (described in the previous step) or for existing users (other than administrative users) on the **General Information** pages of users. If a home mashup is not set for a general user, that user is redirected to the default ThingWorx search page.

6. Click **Save**.

Add Servlet Filters to Tomcat Configuration

Using the Windchill IdP authentication filter, the ThingWorx unauthenticated user is redirected to the Windchill login form for authentication credentials. After successful authentication, the ThingWorx application receives a key and user name. For more information, see Configuring ThingWorx Single Sign-On when Connecting to Windchill on page 33.

The filter is configured on the ThingWorx side in the `web.xml` file that is under the ThingWorx Tomcat installation directory. Use the following steps:

1. Stop Tomcat.
2. Copy `ptc-identity-provider-authentication-filter-{version}.jar` to the `WEB-INF/lib` directory that is under the ThingWorx Tomcat installation directory.

> 📋 **Note**
>
> The JAR file is included with the extension files in `Framework.zip`.

3. Navigate to the `web.xml` file in the following location: `<Tomcat Install Location>\Apache Software Foundation\Tomcat 8.0\webapps\Thingworx\WEB-INF\web.xml`.
4. Add code blocks to the ThingWorx `web.xml` file.

> 📋 **Note**
>
> Note the following about code examples:
>
> - Code examples in this guide may have been reformatted to fit on the page and, therefore, may contain line numbers, hidden editing characters (such as tabs and end-of-line characters) and extraneous spaces. If you cut and paste code from this manual, check for these characters and remove them before attempting to use the example in your application.
> - You must include actual values. Within each code block, replace the content identified by `[]` with values for your implementation.

a. Before the `AuthenticationFilter` block, add the following code block:

```
<filter>
    <filter-name>IdentityProviderAuthenticationFilter</filter-name>
    <filter-class>com.ptc.connected.plm.thingworx.wc.idp.client.filter.
        IdentityProviderAuthenticationFilter</filter-class>
    <init-param>
        <param-name>idpLoginUrl</param-name>
         <param-value>[http or https]://[windchill-host]:[windchill-port]/
            [windchill-web-app]/wtcore/jsp/genIdKey.jsp</param-value>
     </init-param>
</filter>
<filter-mapping>
    <filter-name>IdentityProviderAuthenticationFilter</filter-name>
    <url-pattern>/*</url-pattern>
 </filter-mapping>
 <filter>
    <filter-name>IdentityProviderKeyValidationFilter</filter-name>
    <filter-class>com.ptc.connected.plm.thingworx.wc.idp.client.filter.
IdentityProviderKeyValidationFilter
    </filter-class>
    <init-param>
        <param-name>keyValidationUrl</param-name>
        <param-value>[http or https]://[windchill-host]:[windchill-port]/
            [wndchill-web-app]/login/validateIdKey.jsp</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>IdentityProviderKeyValidationFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

b. At the end of all filter mappings, add the following code block as the last filter in the filter chain:

```
<filter>
     <filter-name>MultiWindchillAuthenticationFilter</filter-name>
     <filter-class>com.ptc.connected.plm.thingworx.wc.idp.client.filter.
        MultiWindchillAuthenticationFilter</filter-class>
    <init-param>
    <param-name>keyGenerationRelativePath</param-name>
    <param-value>wtcore/jsp/genIdKey.jsp</param-value>
    </init-param>
    <init-param>
    <param-name>keyValidationRelativePath</param-name>
    <param-value>login/validateIdKey.jsp</param-value>
     </init-param>
    <init-param>
    <param-name>mwaTimePeriod</param-name>
    <param-value>-1</param-value>
    </init-param>
    <init-param>
    <param-name>excludedExtensions</param-name>
    <param-value>css,js</param-value>
     </init-param>
     <init-param>
    <param-name>pingRelativePath</param-name>
    <param-value>servlet/WindchillGW/wt.httpgw.HTTPServer/echo</param-
value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>MultiWindchillAuthenticationFilter</filter-name>
```

```
        <url-pattern>/*</url-pattern>
    </filter-mapping>
```

5. Restart Tomcat.

Modify ThingWorx Run Time Permissions

Permissions for Administrators

Administrative users need to be added to the ThingWorx Administrators group. This group is a systems object, not visible in the **Groups** menu by default. To access the group, enter `Administrators` in the spotlight search at the top of the ThingWorx screen.

Permissions for Users

By default, permissions for ThingWorx non-administrative users are set to read only. Determine how you will manage your user access and modify the permissions in the **Run Time** or **Run Time Instance** permissions for entities. For example, you can change the access to the **ptc-windchill-demo-thing** to allow read, write, and execute permissions.

Verifying Single Sign On

Both Windchill and ThingWorx need to agree on the name of the administrator user. Consider the following options:

- If you have not made changes in Windchill, an administrator user named "Administrator" was created when Windchill was installed. Using a ThingWorx system configured for single sign-on, you can authenticate as that user and have full access rights as the administrator user in ThingWorx.

- If you have modified Windchill so that there is not a user named "Administrator," then you must decide on a username that is common to both Windchill and ThingWorx and add that user to ThingWorx and the Administrators user group.

To verify the single sign-on configuration, open a browser to the ThingWorx URL. The browser is routed to Windchill for authentication. Provide the Windchill credentials for Administrator (or another user configured to be the ThingWorx administrator). The browser is routed back to ThingWorx, which will open the ThingWorx Composer. Verify you are now running in ThingWorx as that user. This verifies that single sign-on is configured properly.

If you have selected to have the authenticator automatically create users, test that next. To verify, open a browser to the ThingWorx URL. You are routed to Windchill for authentication. Provide Windchill credentials of a user that does not exist in ThingWorx. Your browser is routed back to ThingWorx on the page specified as the home mashup. Verify you are now running as the correct user in ThingWorx.

4

Using the Sample Mashups

This chapter describes the product structure and search sample mashups that are installed with the PTC Windchill Extension. The sections to follow describe how to configure, authenticate, and reuse the sample mashups that have been provided.

Configuring Sample Mashups

To view sample mashups, select **Mashups** under **VISUALIZATION** in the ThingWorx Composer main navigation pane:
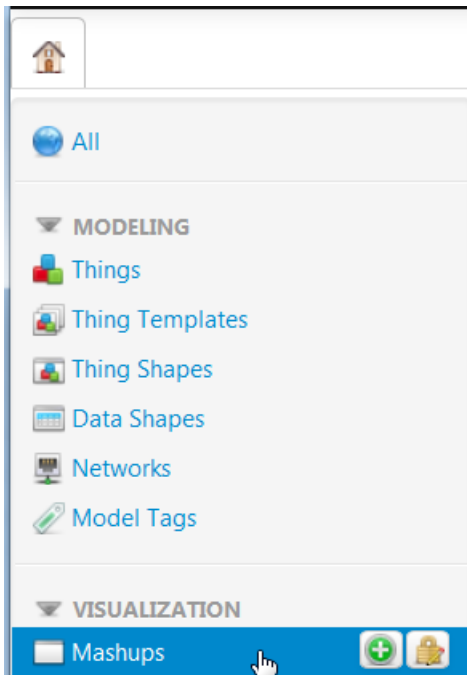
Two sample mashups are included with this release of PTC Windchill Extension:

- **ptc-connected-demo-mashup**

  Product structure sample mashup. For more information, see Product Structure Sample Mashup on page 43.

- **ptc-search-demo-mashup**

  Search sample mashup. For more information, see Search Sample Mashup on page 48.

The sample mashups use the services from the **ptc-windchill-demo-thing** entity that was created from the WindchillConnector Thing Template. While ideal for demo environments, you should not use the **ptc-windchill-demo-thing** in your production mashups. Rather, create a new Thing from the WindchillConnector.

Connecting Mashups to Windchill

To use both mashups, you must set up a connection to a Windchill instance. The connection is stored as the base URL in the **ptc-windchill-demo-thing** entity. From the ThingWorx Composer, complete the following:

1. From the navigation pane on the left, select **Things** under **MODELING**.

2. Edit the **ptc-windchill-demo-thing**.

3. Select **Configuration** under **ENTITY INFORMATION**.
4. Set the **baseURL** value and click **Save**.

    For example:

| baseURL	http://localhost:7080/Windchill

Product Structure Sample Mashup

The product structure mashup retrieves a Windchill product structure based on the input of a Windchill part, Windchill document, or EPM (CAD) document number. The structure table that is returned also emulates the inclusion of data coming from an ERP system.

Before using the mashup, you must and gather one or more valid numbers that you can use to provide data in the sample mashup.

After you have configured the product structure sample mashup to connect to your Windchill instance, you can display the mashup in a browser by clicking **View Mashup**.
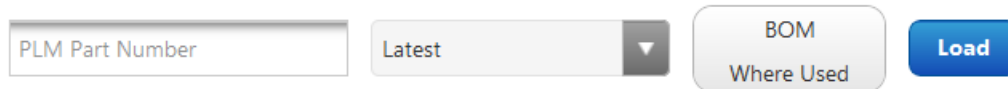
When you initially display the sample mashup, the following opens in your browser:
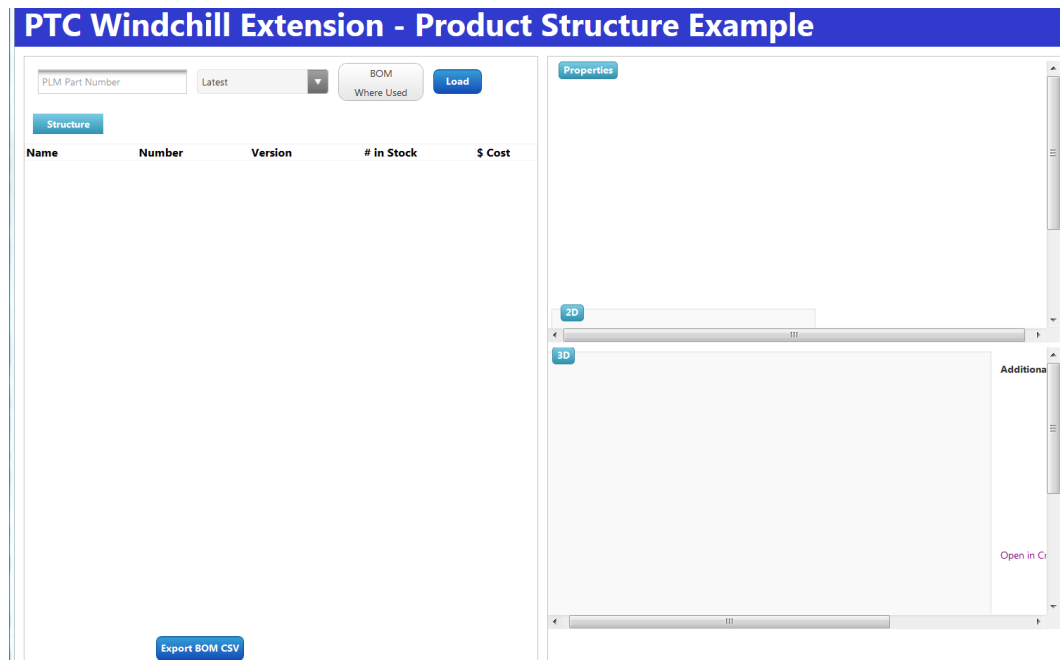
The product structure sample mashup consists of four main areas as described in the following sections.

Input Field and Widgets

The following product structure sample mashup screen shows the initial input area that is located on the top left side of the window:

The initial input area contains the following:

- An input field for entering a Windchill part, Windchill document, or EPM document number.

  In the input field, you enter a valid number for a Windchill part, Windchill document, or EPM document (also known as CAD document) that is available from the Windchill instance to which your application is connected. The user under which you are logged in must have the correct Windchill access permissions that allows you to view the object represented by the number you enter.

- The following widgets can be used to affect what is loaded:

  ○ The drop-down widget containing **Latest**.

    The sample mashup is set up to only display the latest version of the object. If you want to view versions other than the latest, you can make modifications to the entity that produces the drop-down widget.

  ○ The split **BOM** / **Where Used** button determines the type of structure displayed in the **Structure** table.

    If neither option is selected, the BOM (bill of materials) for the object is returned.

    For more information on the **Structure** table, see the sections that follow: Structure Table for BOM and Structure Table for Where Used.

- The **Load** button.

  Clicking **Load** initiates the loading of the **Structure** table.

  If it takes too long to load or you do not have the required permissions, the following image displays:

Structure Table for BOM

On the left side of the window, enter a valid part number and click **Load** to populate the **Structure** table. Optionally, you can also click **BOM** before clicking **Load** for the same result.

Valid part numbers come from the data that is available on your Windchill system. The following screen shows the result of entering the top-level part number `WCDS000594` and clicking **Load**:

Name	Number	Version	# in Stock	$ Cost
▼ 01-2_engine.asm	WCDS000594	A.1 (Design)	435	654.00
▼ tensioner.asm	WCDS000601	A.1 (Design)	427	1203.00
tensioner_bracket.prt	WCDS000233	A.1 (Design)	404	1315.00
tensioner_guide.prt	WCDS000109	A.1 (Design)	263	1306.00
power_steering_pump.prt	WCDS000525	A.1 (Design)	581	122.00
01-2-timing-cover.prt	WCDS000474	A.1 (Design)	244	1385.00
▼ oil_pan.asm	WCDS000432	A.1 (Design)	71	751.00
oil_pan.prt	WCDS000760	A.1 (Design)	251	199.00
▶ oilpump.asm	WCDS000645	A.1 (Design)	975	1283.00
▼ 01-2_head.asm	WCDS000159	A.1 (Design)	275	356.00
▶ 01-2_cam_exhaust.asm	WCDS000494	A.1 (Design)	540	869.00
▶ 01-2_valve_intake_2.asm	WCDS000052	A.1 (Design)	719	1343.00
▶ 01-2_valve_intake_2.asm	WCDS000052	A.1 (Design)	719	1343.00
▶ 01-2_valve_intake_2.asm	WCDS000052	A.1 (Design)	719	1343.00
▶ 01-2_valve_intake_2.asm	WCDS000052	A.1 (Design)	719	1343.00
01-2_front_oil_seal.prt	WCDS000298	A.1 (Design)	272	1342.00
01-2_front_oil_seal.prt	WCDS000298	A.1 (Design)	272	1342.00
01-2_chain.prt	WCDS000646	A.1 (Design)	7	626.00
▶ valve_exhaust_2.asm	WCDS000454	A.1 (Design)	425	150.00
▶ valve_exhaust_2.asm	WCDS000454	A.1 (Design)	425	150.00
▶ valve_exhaust_2.asm	WCDS000454	A.1 (Design)	425	150.00
▶ valve_exhaust_2.asm	WCDS000454	A.1 (Design)	425	150.00
▶ 01-2_cam_intake.asm	WCDS000658	A.1 (Design)	267	136.00
01-2_cam_bracket.prt	WCDS000656	A.1 (Design)	369	75.00
01-2_cam_bracket.prt	WCDS000656	A.1 (Design)	369	75.00

The rows in the table show the nested parts under the top-level part that was entered. The **Name**, **Number**, and **Version** columns are populated by the data coming from Windchill. The values in the additional **# in Stock** and **$ Cost** columns emulate ERP data that could be added to the structure table.
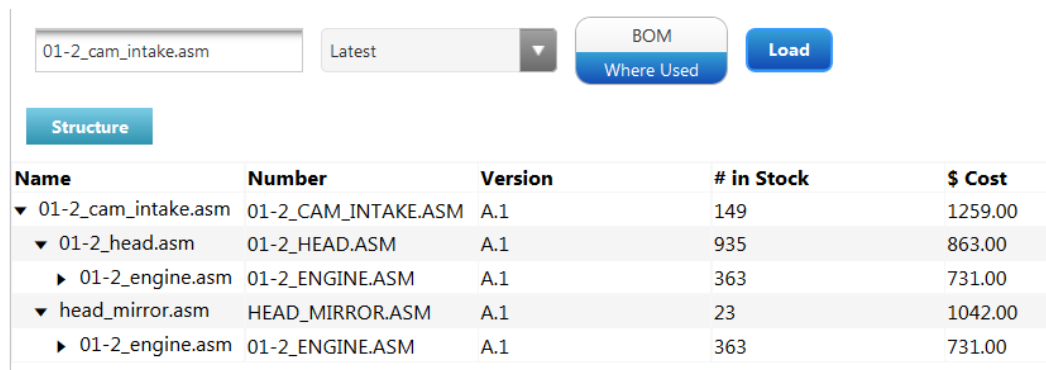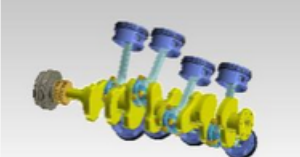
📝 Note

The data included in the sample does not come from an ERP system; it is data generated through Java script that is included with the sample mashup.

Clicking **Export BOM CSV** exports the BOM data for the parts in the **Structure** table into a CSV file, allowing you to either open the file in Excel or save the file. The BOM data includes all data column values that are stored in Windchill. All columns included in the Data Shape associated with the binding service of widgets are exported.

Structure Table for Where Used

On the left side of the window, enter a valid number (such as a CAD document number), click **Where Used**, and then click **Load** to populate the **Structure** table. The following screen shows the where used result of entering the CAD document number `01-2_CAM_INTAKE.ASM`:

The rows in the table show the assemblies that use the intake assembly. The **Name**, **Number**, and **Version** columns are populated by the data coming from Windchill. The values in the additional **# in Stock** and **$ Cost** columns emulate ERP data that could be added to the structure table.

📝 **Note**

The data included in the sample does not come from an ERP system; it is data generated through Java script that is included with the sample mashup.

Clicking **Export BOM CSV** exports the data displayed in the **Structure** table into a CSV file, allowing you to either open the file in Excel or save the file. The where-used data includes all data column values that are stored in Windchill. All columns included in the Data Shape associated with the binding service of widgets are exported.

Properties and 2D View

The **Properties** and **2D** widgets on the upper right side of the window are populated when you select a part from the **Structure** table. For example, if you select 01-2_crankshaft.asm, the following screen shows an example of the **Properties** and **2D** widgets for the part:

name	01-2_crankshaft.asm
number	WCDS000204
objectId	OR:wt.part.WTPart:64777
parentId	/
quantityInStock	204
state	In Work
treeId	/481
type	Part

Next to the 2D drawing are two buttons:

- **Export Properties CSV** saves the properties that are currently displayed in a CSV file, allowing you to either open the file in Excel or save the file.
- **Create Problem Report** creates a problem report for the part that is selected in the **Structure** table. After the report is created, the window updates to include a link to the Windchill problem report page.

3D View

The **3D** widget on the lower right side of the window displays 3D data that includes the selected part. This widget is part of the PTC Creo View Extension on page . For example, the crankshaft shown in the previous section is included in the following 3D view:
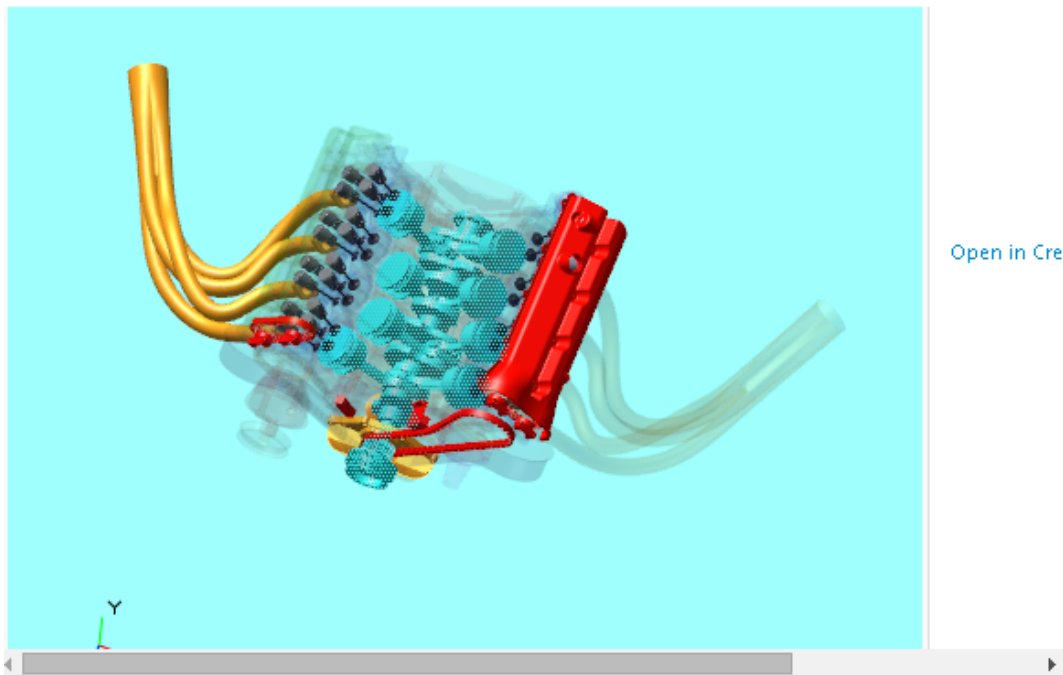
> **📝 Note**
>
> 3D images display on computers that have PTC Creo View installed.

The **Open in Creo View** link to the right of the 3D data allows you to open the 3D data in the Creo View product.

For Creo View product requirements, see Product Requirements on page 8.

Search Sample Mashup

PTC provides a search sample mashup that you can view in the ThingWorx composer to demonstrate the use of Windchill search capabilities. Before using the mashup, you must configure a Windchill connection and have one or more saved searches in the Windchill instance.

After you have configured the search sample mashup to connect to your Windchill instance, you can display the mashup in a browser by clicking **View Mashup**.

When you initially display the sample mashup, the following opens in your browser:

Search Input Options

The initial input area is located on the top left side of the window and contains the following options:

- **Saved Search** list

  This list contains the searches that have been saved as a global search in Windchill. Application developers can predefine the search scope and then only expose the search keyword field in an application (rather than also providing access to the saved search list).

- **Keyword** field (optional)

  Entering a keyword filters the saved search results by the keyword. Any keywords you enter replace keywords that were included in the Windchill saved search. Question mark (?) and asterisk (*) wildcards can be used in this field,

- **Search** button

  Clicking **Search** initiates the loading of the **Search Results** table.

  If it takes too long to load or you do not have the required permissions, the following image appears:

Search Results Table

The following screen shows the search results from a saved search for parts with a number that begins with "W."

Search Results

| Name	Number	Type	Version	State	DefaultUnit	
01-51328_outer_race.prt	WCDS000100	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64701
01-52003.prt	WCDS000110	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64700
oilpump_out_ring.prt	WCDS000591	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64703
01-51373.prt	WCDS000427	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64702
01-2_cam_exhaust.asm	WCDS000494	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64697
01-51240.prt	WCDS000478	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64696
01-51361.prt	WCDS000554	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64699
01-31002.prt	WCDS000317	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64698
01-51368d.prt	WCDS000023	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64693
01-2_piston_pin.prt	WCDS000045	Part	A.1 (Design)	In Work	each	OR:wt.part.WTPart:64695

The rows in the table include the **Name**, **Number**, **Type**, **Version**, **State**, **DefaultUnit**, and **ObjectId** columns which are populated by the data coming from Windchill.

Reusing Sample Mashup Content

After you have the Windchill connection established for the sample mashups and have the mashups working with your data, PTC recommends that you create duplicate mashups that you can change to meet your needs.

1. In the ThingWorx Composer, select one of the sample mashups and then click **Duplicate**.
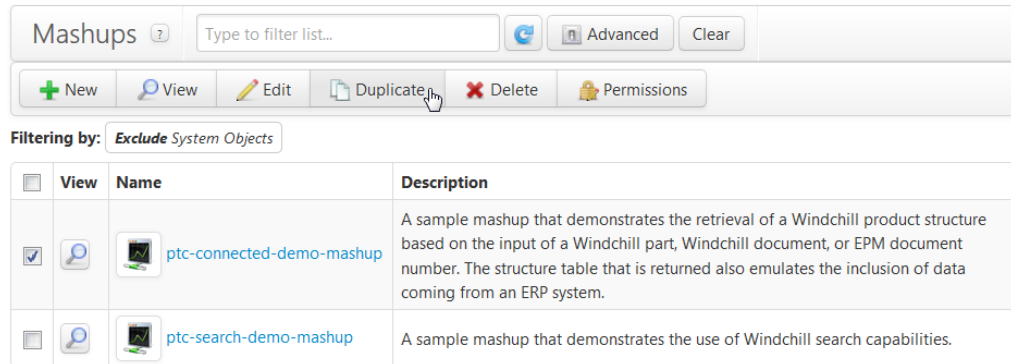

2. From the new mashup that is created, enter a new name and description and save the mashup.
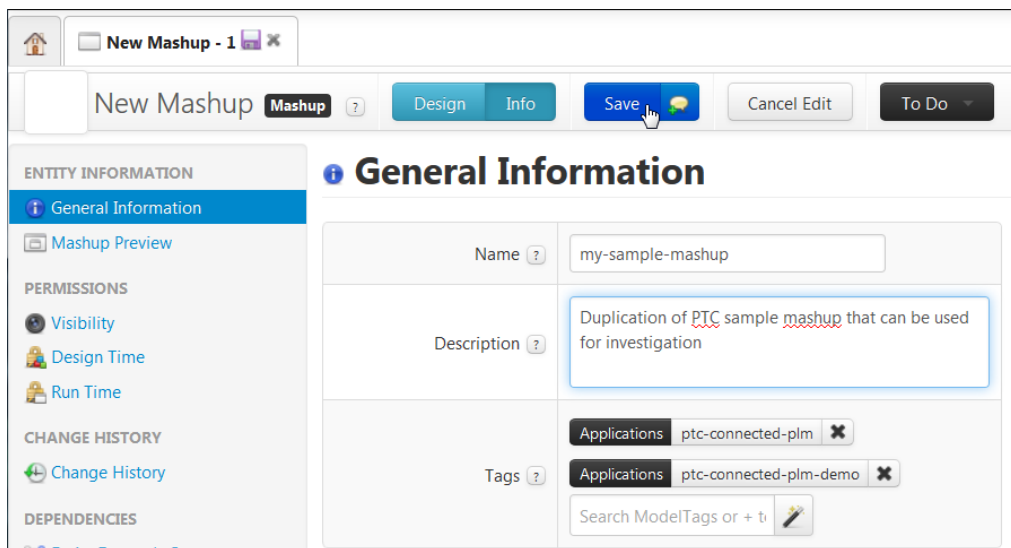
After the mashup is saved, you can edit your copy. To learn more about the entities in the product structure mashup, view the **PTC Windchill Extension Sample Mashup for ThingWorx** video:

http://youtu.be/P8hLk5_pRl0

Additionally, you can inspect the entities and services in the mashup. Clicking items in the main display panel changes what you see on secondary panels. For example, the following screen shows the connections and properties for the **Export BOM CSV** button:
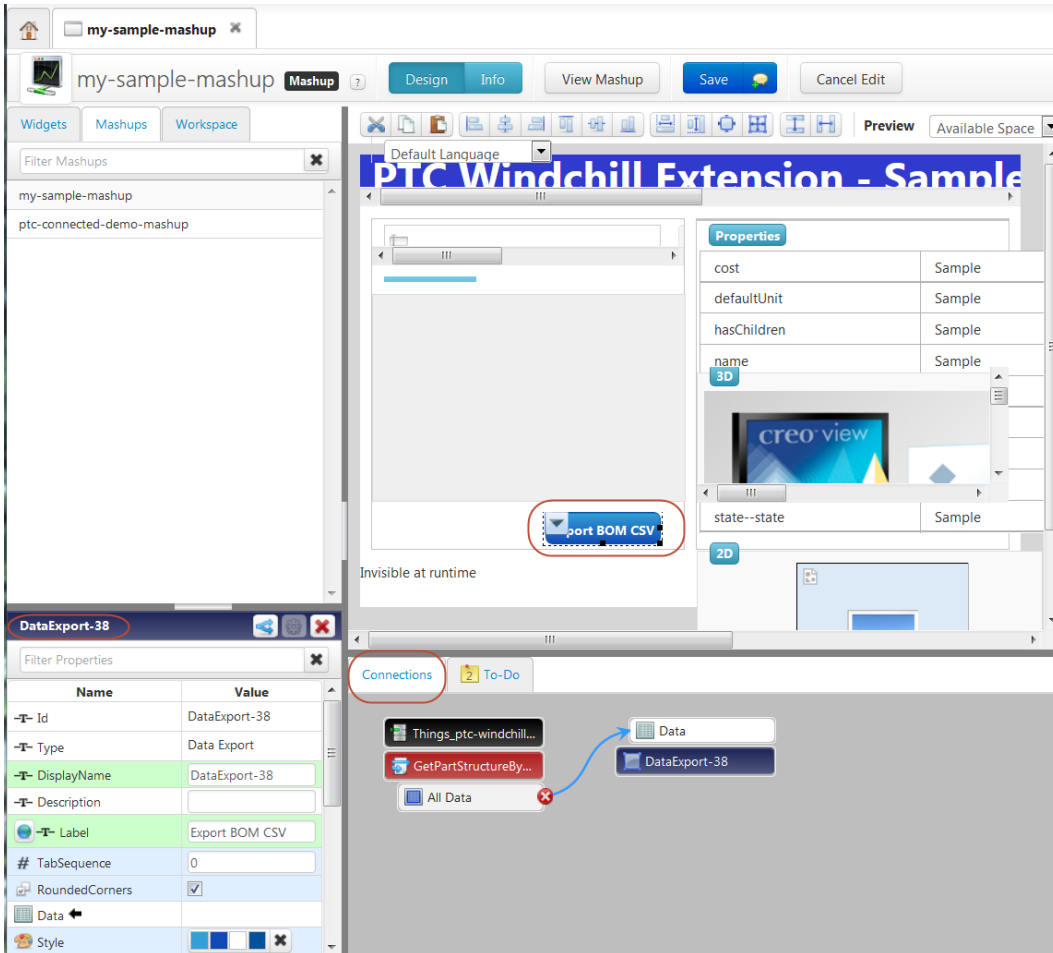

For details on specific entities, look for **Description** fields, hover over names, and click help icons.

The following shows an example of clicking a help icon:
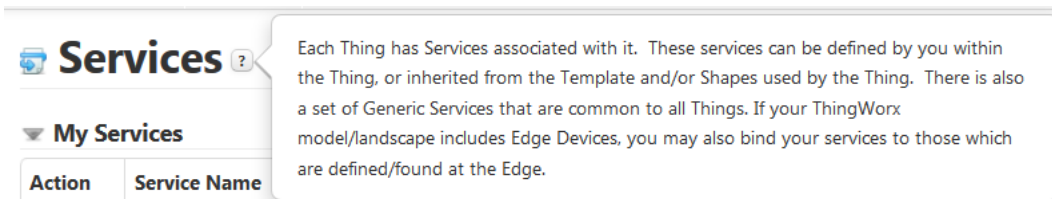
The following shows an example of hovering over a service name:

The following sections show common activities that could be useful as you edit your copy of the sample mashup.

Changing Number of Structure Levels in Sample Mashup

In the product structure sample mashup, the **GetPartStructureByNumber** service, available from **ptc-windchill-demo-thing**, controls the number of nested part levels in the **Structure** table.
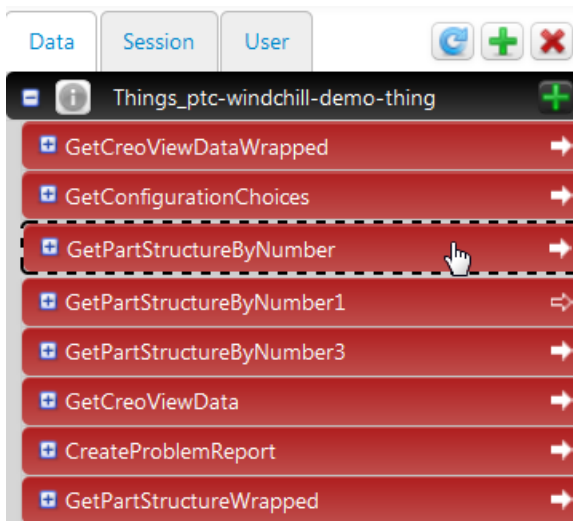
Clicking the service updates the properties shown below the services. The default levels value is 3. To change this value, modify the **123 levels** field.
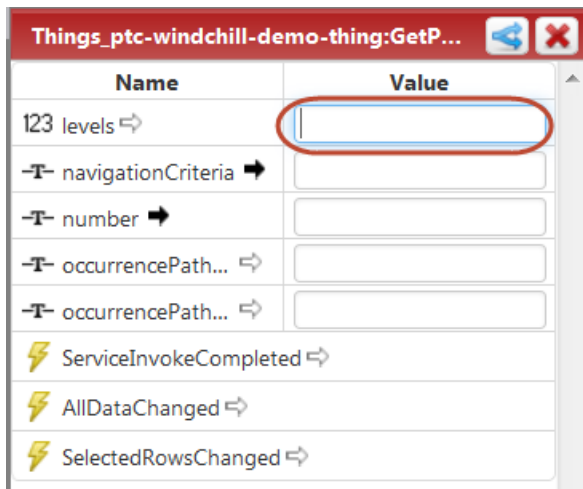
For example, entering 1 for **123 levels** restricts the expansion in the **Structure** table to one level below the part entered.

Changing Number of Auto-Expand Levels in Sample Mashup

In the product structure sample mashup, the tree table used to display parts has properties that are defined for the table. To display those properties when editing your copy of the sample mashup, click the tree table:
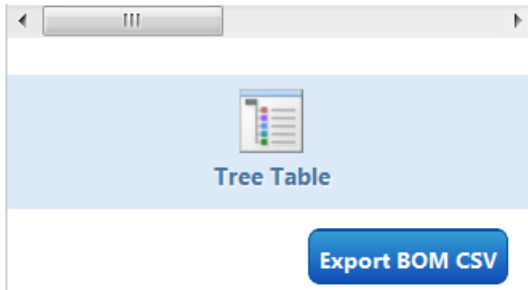
The table properties display in the lower left corner of the screen:

The **CollapseChildernOfLevel** property identifies how many levels are expanded when you initially load part data. The value you supply for this property must be less than or equal to the number of levels specified when a BOM is loaded. For details on setting the levels of part data loaded, see Changing Number of Structure Levels in Sample Mashup on page 54.

Scroll to display the **CollapseChildernOfLevel** property and enter a new value:
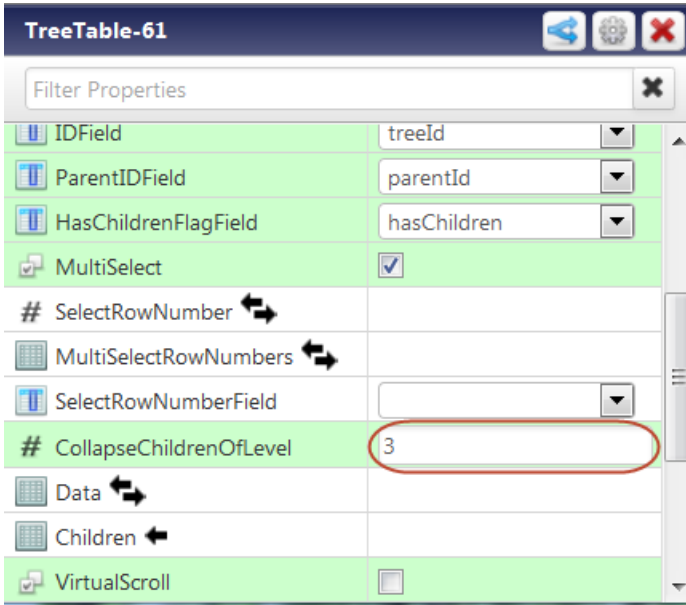
💡 **Tip**

By hovering over the properties in the tree table, you can see descriptions for each property.

PTC Windchill® Extension Guide
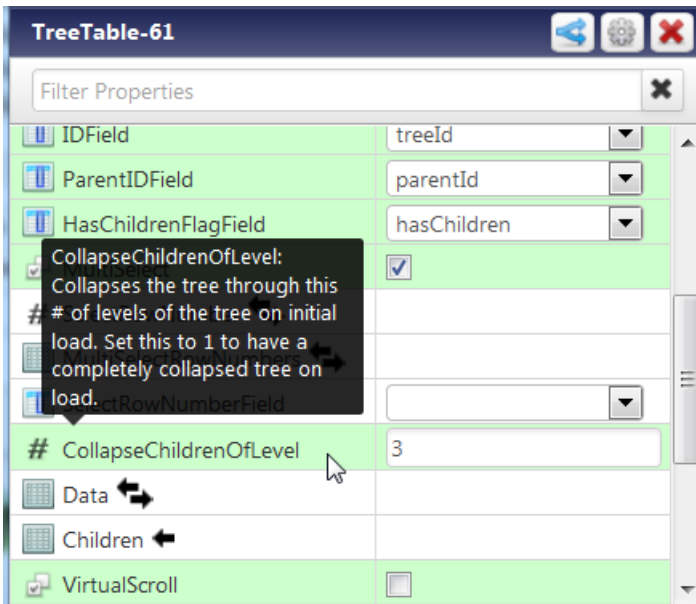
5

Developing a Customized Application

This chapter provides additional information on ways you can develop a ThingWorx application that connects to Windchill.

Localizing Labels and Fields in Your Application

ThingWorx supports the localization of the runtime user interface of your ThingWorx application. This includes being able to localize elements, such as the labels and fields, that you create when using PTC Windchill Extension entities.

To localize elements in entities, you must include a token for each element. Click the localization toggle icon to search for and select an existing token or create a new token for a field.
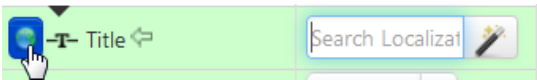
All tokens that you create are added to the localization tables maintained by ThingWorx. By creating and populating localization tables, you can display runtime elements in different languages. For details on how to create and populate localization tables, search for "localization tables" in the ThingWorx help center. The help center is available from the PTC Help Centers page.

For more information on how to localize labels, see the following tutorial video:

http://youtu.be/3bo6HtXSqRA

To provide localized text for a large number of elements, export a localization table, add localized text to the exported XML file, and import the file.

Adding Saved Filters to Your Mashup

In Windchill, you can create and save filters for parts, documents, or EPM (CAD) documents. You can use these saved filters to display filtered structures in a ThingWorx mashup.

Creating and saving filters for parts, documents, or CAD documents can be done from the corresponding information page of the object. To use saved filters in a ThingWorx application, follow these best practices:

- In a part structure mashup, a filter can only be specified by name. Ensure that all filters that you want to use in a ThingWorx application have unique names.

- In most instances, saved filters are intended to be used with the specific object type for which they were created. Ensure the selected filters work with the object structure you want to display in the ThingWorx application.

PTC Navigate

Saved Filter options are available on the app tailoring pages. For more information, see the "Basic App Tailoring" section.

PTC Windchill Extension

You can code your ThingWorx Composer application to prompt the user for the saved filter. For example, the part structure sample mashup code provided by PTC includes a drop-down widget next to the **PLM Part Number** field that has one item (**Latest**):

You can modify the code for this drop-down widget to list any number of saved filters.

The output of the **GetConfigurationChoices** service is bound to the input of this drop-down widget. **GetConfigurationChoices** is a service of the **ptc-windchill-demo-thing**. You can edit the **GetConfigurationChoices** service script to modify the items that are shown in the drop-down widget to include one or more saved filter names. When selected, a saved filter is passed to Windchill.

The **GetConfigurationChoices** service returns an Infotable with two fields:

- `label` is the value shown in the drop-down widget.
- `value` is the name of the saved filter that is passed to Windchill.

Using PTC Creo View Extension

The PTC Creo View Extension, included with the PTC Windchill Extension, enables you to visualize 3D data from Windchill within a ThingWorx application. The PTC Creo View Extension provides the following capabilities:

- Interact with the PTC Creo View 3D data from Windchill. For example, you can zoom, pan, and rotate the 3D graphics.
- Trigger events when PTC Creo View parts are selected.
- Change the transparency and color of the PTC Creo View parts.
- Change the background color of the PTC Creo View plugin.

Properties and Events

PTC Creo View Extension has the following properties and events:

- `ProductToView` – This property enables you to visualize 3D data from Windchill in the PTC Creo View Extension. It is the URL to the PTC Creo View data file. Populate the `ProductToView` property to visualize the data.

Windchill sample mashups provided with the installation demonstrate how to use sample services to populate this property.

- `BackgroundStyle` – This property enables you to change the background color of the PTC Creo View 3D graphics area.

- `Data` – This property enables you to select PTC Creo View parts, and also change their color and transparency. You must bind this property to a data table. The data table must contain a column which specifies the Occurrence ID of the part. This is the unique ID of the part in the PTC Creo View 3D data.

- `DataFormatter` – This property enables you to specify rules that must be used to apply color and transparency to the PTC Creo View parts. Specify a column for these rules from the data table which is bound to the `Data` property.

- `OccurrenceField` – This property enables you to select the column that must be used as the Occurrence ID. The column is selected from the data table which is bound to the `Data` property.

- `selectedOccurrencePath` – This property specifies the Occurrence ID of the part selected in the 3D data.

- `selectionChanged` – This event indicates a change in the selection of 3D data.

Using Info*Engine Tasks when Developing Your Application

PTC includes the general-purpose **ExecuteTask** service in the **WindchillConnector** thing template. The **ExecuteTask** service allows you to invoke any task in the Windchill task codebase.

- Specify the task to invoke by URI.
- Define the input parameters.
- Optional: Define an Infotable where row data is converted into Info*Engine group elements and is used as the task's input group.

To use the **ExecuteTask** service, create a thing using the **WindchillConnector** thing template. For example, you can add a new service in **ptc-windchill-demo-thing** as shown in the following screen.
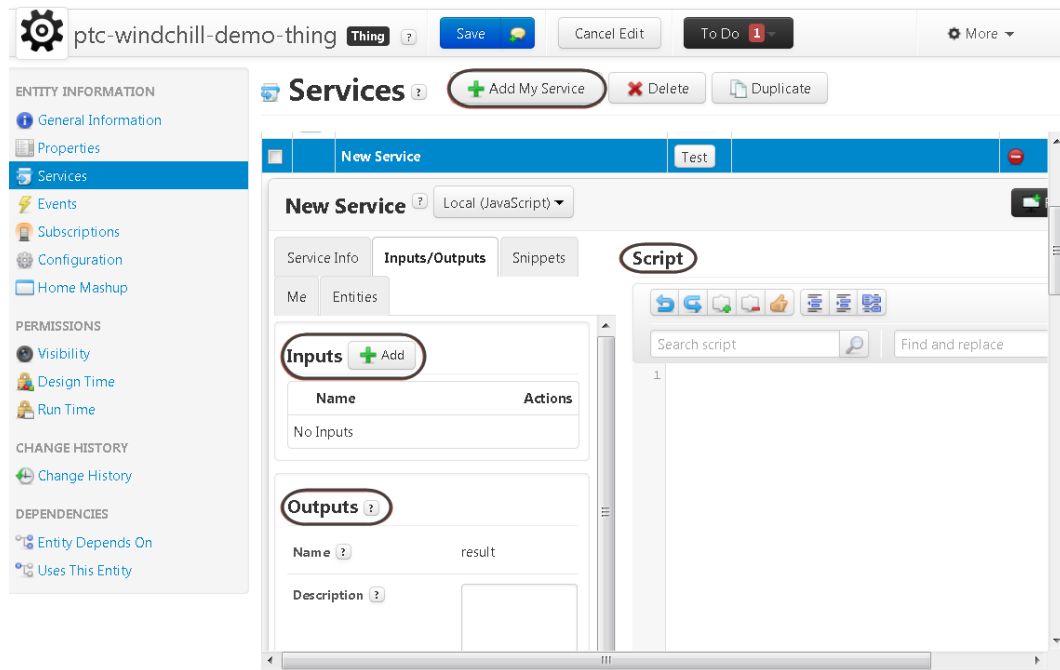
The following sections describe some common application features that you can include using Info*Engine tasks: and .

Querying WTParts By Name

This example calls the **ExecuteTask** service that is provided in the **WindchillConnector** thing template to query parts by name.

The example assumes that you have already created a compatible data shape that represents WTParts. It also assumes that you are implementing a new service from within the thing that is used to execute the query. Additionally, you have created the following Info*Engine task and have stored the task under the Windchill installation directory:

```
$WT_HOME/tasks/com/ptc/windchill/example/
queryPartByName.xml
```

```
<%@page language="java" access="http"%>
<%@taglib uri="http://www.ptc.com/infoengine/taglib/core" prefix="ie"%>

<ie:webject name="Query-Objects" type="OBJ">
  <ie:param name="INSTANCE" data="$(@FORM[]supporting-adapter[*])"
        delim="!" valueSeparator="!"
```

```
        default="<%=com.infoengine.au.NamingService.getVMName()%>"/>
  <ie:param name="TYPE" data="wt.part.WTPart"/>
  <ie:param name="WHERE" data="name = '${@FORM[]name[]}'"/>
</ie:webject>
```

Complete the following steps to create a ThingWorx service that can be used to query for parts by name. Use the thing you have created that uses the **WindchillConnector** thing template.

1. From the newly added service, enter service information (such as name and description) and the input and output parameters:

   - **Input Parameter**:

     STRING name

     Input Parameter

Name ?	name
Description ?	
Base Type ?	-T- STRING ▼
Required ?	✓
Has Default Value ?	☐

   - **Outputs**:

     Base Type: INFOTABLE

     Data Shape: my.wt.part.WTPart

Replace **Data Shape** value with the name of your data shape:

2. Enter the following script that executes the `queryPartByName.xml` task:

> 📋 **Note**
>
> Code examples have been reformatted to fit the page and could contain line numbers, hidden editing characters (such as tabs and end-of-line characters), and extraneous spaces. If you cut and paste code, check for these characters and remove them before attempting to use the example in your application.

```
/*
 * The ExecuteTask service takes as arguments an InfoTable of IEProperty
 * DataShape defining the tasks parameters, a String defining the URI of
 * the task to execute, and an optional InfoTable defining objects to use as
 * the tasks input group. For this task, define the parameters and the task URI
 */

/* construct the InfoTable defining the parameters */
var params = {
  infoTableName : "InfoTable",
  dataShapeName : "IEProperty"
};
var inputParams = Resources["InfoTableFunctions"].
CreateInfoTableFromDataShape(params);

/* IEProperty DataShape has the fields name and value */
inputParams.AddRow({
  name: 'name', //this is the name of the parameter
  value: name //and it's value - which is the argument to this service
});

/* now call the ExecuteTask service */
var params = {
  params: inputParams,
  dataShape: 'my.wt.part.WTPart', //replace with name of your data shape
  uri: 'com/ptc/windchill/example/queryPartByName.xml', //task to invoke
  objects: undefined //task does not take input group of objects; leave undefined
};
var result = me.ExecuteTask(params);
```

3. Save the service.

Using Custom Reports from Windchill

📋 Note

For more information on custom reports, see the PTC Windchill Help Center or *PTC Windchill Customization Guide* available on ptc.com.

Overview

This example also calls the **ExecuteTask** service that is provided in the **WindchillConnector** thing template. It uses the Info*Engine task `ExecuteReportTemplate` to execute a custom report. In this example, you create a new service method that executes the report and returns the report as an Infotable. At a high level, here is what the new service method does:

1. Use the **WindchillConnector Query** service to query for the context in which the report template lives.
2. Use the **WindchillConnector Query** service to query for the report template of the report to execute.
3. Use the **WindchillConnector ExecuteTask** service to execute the Info*Engine task named `ExecuteReportTemplate`, which generates the report results.

Before You Begin

Before you can write this service, create the data shapes that this service needs.

1. Use the **WindchillConnector CreateDataShapes** service to create the shape for the Windchill context object whose type is `wt.inf.container.WTContainer`.
2. Use the **WindchillConnector CreateDataShapes** service to create the shape for the Windchill report template object whose type is `wt.query.template.ReportTemplate`.

📋 Note

Info*Engine cannot retrieve the report template object because it contains BLOBed XML data. Remove any fields from your data shape that would contain XML or QML data.

---

3. Create a data shape that matches the fields of the custom report.

Implementing the Service

Now you can implement the service that executes the custom report and returns the report as an Info Table.

1. Query for the context in which the report template lives. For example, this code queries for the "Drive System" context:

```
var containers = Things["ptc-windchill-demo-thing"].Query({
    criteria:"name = 'Drive System'",
    dataShape: 'demo.wt.inf.container.WTContainer',
    type: 'WCTYPE|wt.inf.container.WTContainer'
});
if(containers.getRowCount() > 0) {
    var containerUfid = containers.getRow(0).ufid;
    logger.debug("Using container UFID " + containerUfid);
}
```

2. Query for the report template by name and its context. For example, this code queries for the report template named "Part Assembly"

```
var reportTemplates = Things["ptc-windchill-demo-thing"].Query({
      criteria: "name = 'Part Assembly'",
    dataShape: 'demo.wt.query.template.ReportTemplate',
      type: 'WCTYPE|wt.query.template.ReportTemplate',
    containerUfid: containerUfid
});
if(reportTemplates.getRowCount() > 0) {
    var reportTemplateUfid = reportTemplates.getRow(0).ufid;
    logger.debug('Using report template UFID ' + reportTemplateUfid);
}
```

3. Invoke the Info*Engine `ExecuteReportTemplate` task. In this example, the report requires parameters "name" and "version," so those are passed as input. Note that the data shape of the resulting report is "AssemblyReport."

```
var properties = Resources["InfoTableFunctions"].
        CreateInfoTableFromDataShape({
    infoTableName : "InfoTable",
    dataShapeName : "IEProperty"
});
properties.AddRow({
    name : 'object_ref',
    value: reportTemplateUfid
});
properties.AddRow({
    name : 'input',
    value: "name='" + partName + "'" + "," + "version='" + partRevision + "'"
});
var result = Things["ptc-windchill-demo-thing"].ExecuteTask({
    params: properties,
    dataShape: 'AssemblyReport',
```

```
        uri: 'com/ptc/windchill/enterprise/report/ExecuteReportTemplate.xml',
        objects: undefined /* INFOTABLE */
    });
```

Modifying Windchill to Use the Service

Before you can use your service, you must modify Windchill so that it allows
HTTP access to the task `ExecuteReportTemplate` task. To do this, add an
entry `access=http` to the `.delegateInfo` files along the path to the task
folder at `tasks\com\ptc\windchill\enterprise\report`.

Using Windchill REST Web Services when Developing Your Application

Windchill provides various REST web services that can be used in a mashup.
These services provide a means to access Windchill data and the processing power
of the Windchill system. The PTC Windchill Extension includes the
WindchillConnector thing template that provides the **processJSONRequest** service
used to invoke Windchill REST web services.

To invoke a Windchill REST service, you can use the **processJSONRequest**
service that is available from **WindchillConnector** thing template. For example, the
following code snippet from **ptc-windchill-demo-thing** invokes the Windchill
service `/structure/objects`, passing parameters that specify the list of
properties, a number, and the `navigationCriteria` name:

```
var params = {
    headers: {
        'Accept': "application/json",
        'Content-Type': 'application/json'
    },
    type: 'GET',
    data: undefined,
    url: me.getRestUrl() + "/structure/objects",
    queryParams: {
      '$select': encodeURIComponent(properties),
      '$filter': "number+eq+'" + encodeURIComponent(number) + "'",
      'navigationCriteria': encodeURIComponent(navigationCriteria)
    },
    timeout: 15000
};
// result: JSON
var jsonResult = me.processJSONRequest(params);
```

Detailed documentation on Windchill REST services is available from the Windchill user interface. The documentation includes descriptions of the resources, endpoints, parameters, and representations and their structure. To access the documentation, click the customization icon in the **Navigator**, and then select **Documentation**.
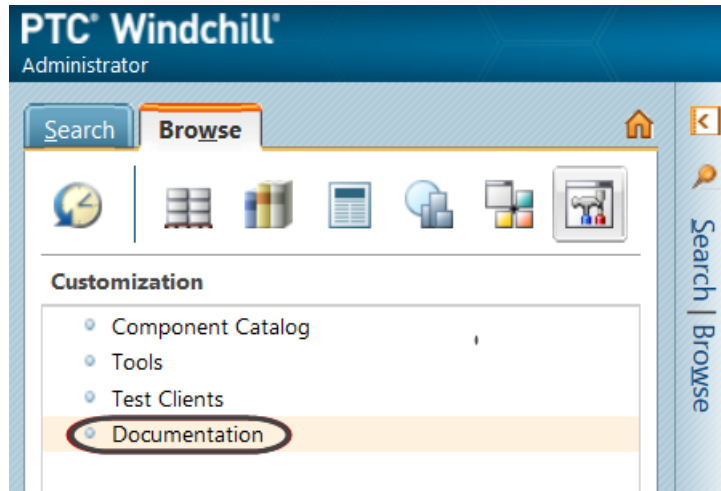
The documentation is available when the **Client Customization** preference in the Windchill user interface is set to **Yes**:

For information on setting the preference, see "Customization Tools Overview" in the PTC Windchill Help Center.

ServiceTimeout Property

The **ServiceTimeout** property controls the default time (in milliseconds) that ThingWorx waits for a REST call to Windchill to complete before Windchill returns an error to the mashup user interface. This default value can be overridden on an individual call by specifying the timeout argument on the call to Windchill. For example, the following code calls the Windchill REST endpoint to execute a saved search and specifies a timeout of 10 seconds. If this call does not specify the timeout argument, then the timeout specified in **ServiceTimeout** is used.

```
var params = {
```

```
queryParams: {
    '$select': 'name,number,version'
},
data: {
    keyword: undefined
},
type: 'POST',
url: me.getRestUrl() + "/search/saved-searches/" + encodeURIComponent(oid) + "/results",
timeout: 10000
};
var jsonResult = me.processJSONRequest(params);
```

The following screen shows ServiceTimeout used in the configuration of demo data:

A

Example Configuration Using SSL for Secure Communications

The following example illustrates one way to configure Windchill and ThingWorx for trusted SSL communications. This example uses a self-signed certificate to secure communications between Windchill and ThingWorx, which is appropriate for environments where the communication occurs within the corporate firewall. Where traffic occurs on the Internet, a commercial trusted certificate is advised.

> **Note**
> * These instructions rely on configuration scripts that come with the most recent version of the PTC HTTP Server. The most recent PTC HTTP Server is included with Windchill 11.0 and the most recent CPS releases for Windchill 10.2.
> * This example assumes that Windchill is already configured to use SSL.
> * The key used here does not need to be the same as that used for either the Windchill or ThingWorx web interface.
> * Code examples have been reformatted to fit the page and could contain line numbers, hidden editing characters (such as tabs and end-of-line characters), and extraneous spaces. If you cut and paste code, check for these characters and remove them before attempting to use the example in your application.

1. Create a truststore for ThingWorx and import the Windchill certificate:
2. Create keystore for ThingWorx.

3. Configure Windchill to trust ThingWorx.

> **📝 Note**
>
> If ThingWorx and Windchill are on separate systems, step 3 a, b, c can be performed on the Windchill server. Then copy the .jks files to the ThingWorx server.

a. Update PTC HTTP Server configuration to reference the CA certificates file.

   i. Navigate to and edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.

   ii. Uncomment the `SSLCACertificateFile` entry so that it refers to the `ca-bundle.crt` file. Note the location of the `ca-bundle.crt` file.

   iii. Save changes.

b. Add the ThingWorx certificate to PTC HTTP Server's list of trusted CA certificates.

   i. Edit the `ca-bundle.crt` file you pointed to previously in the `SSLCACertificateFile` entry.

   > **📝 Note**
   >
   > If the file does not yet exist, create the file and any required directories.

   ii. Append the contents of the ThingWorx certificate (servercert.pem from step 2 a above) held in the trust store to this file.

   iii. Save changes.

c. Configure PTC HTTP Server to export the certificate information to Windchill.

   i. Edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.

   ii. Find the line with `SSLOptions`.

   iii. Uncomment out the line with `SSLOptions +ExportCertData`.

      For example:

      ```
      SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
      <FilesMatch "\.(cgi|shtml|phtml|php)$">
      ```

```
        SSLOptions +StdEnvVars
/FilesMatch>
<Directory "C:/ptc/Windchill_10.2/HTTPServer/cgi-bin">
        SSLOptions +StdEnvVars
</Directory>
```

    iv. Save changes.

  d. Configure PTC HTTP Server for the sslClientAuth URL.

    i. Find the Windchill web app name in `<windchill-home>/ codebase/wt.properties` in the entry `wt.webapp.name`.

    ii. Open a shell or command prompt and enter the following:

```
cd <HTTPSERVER_HOME>ant -f webAppConfig.xml -DappName=[windchill-web-app] -Dresou
```

    iii. Configure Windchill to trust the ThingWorx certificate.

       1) Edit `<windchill-home>/codebase/WEB-INF/web.xml`.

       2) Find `<filter-name>TrustedSSLAuthFilter</ filtername>`.

       3) Add another *<init-param>* element after the existing one where *[thingworx-cert-name]* is the name of the ThingWorx certificate. For a self-signed certificate, the certificate name is the Owner value for the certificate when running `keytool -list -v -keystore <path-to-ThingWorxKeyStore.jks>`:

```
<init-param><param-name>trustedSubjectPattern.1</param-name><param-value>[thi
```

    iv. Save changes.

4. Configure ThingWorx for SSL.

  a. In ThingWorx Composer, edit the **WindchillConnector** thing template that represents the connection to the Windchill system being configured.

  b. Specify path to `ThingWorxKeyStore.jks` and also specify the password.

  c. Specify path to `ThingWorxTrustStore.jks` and also specify the password.

  d. Specify the Windchill `https` address and SSL port.

  e. Save the thing.

  f. View the ThingWorx Application Log to verify that no configuration errors exist.

5. Launch **ptc-connected-demo-mashup** to verify the ThingWorx – Windchill trusted SSL configuration.

B

Architecture of Single Sign-On

Use the authentication filter and authenticator extension provided by PTC to implement a single sign-on for users accessing a ThingWorx application that is connected to Windchill.

In this overview, the Windchill server is the remote IdP. The authentication filter and authenticator extension are installed in ThingWorx.

The architecture consists of two components:

- PTC Identity Provider Authentication Filter – Tomcat filter for authentication; the filter redirects unauthenticated users to the IdP and validates the key.

- PTC Identity Provider Authenticator Extension – ThingWorx extension for authentication; the extension includes dynamic user creation and setting home mashup.

Design of Authentication Filter

The purpose of this filter is to delegate the ThingWorx authentication to the remote authentication service. The authentication filter follows the standard single sign-on architecture.

The following diagram shows the authentication flow:
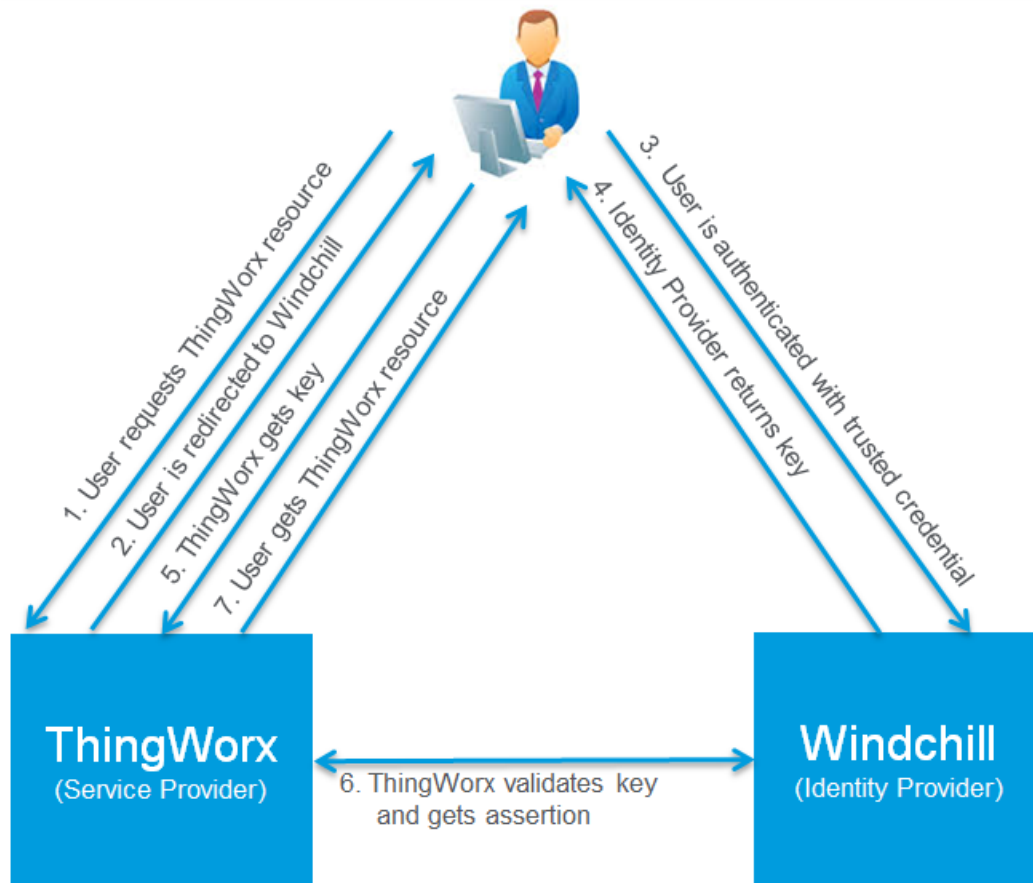
The following authentication steps expand on the steps shown in the diagram. Each numbered step corresponds with the same number in the diagram:

1. User attempts to access the ThingWorx application.

2. The PTC Identity Provider Authentication Filter intercepts the ThingWorx user request and redirects it to IdP for login.

   User sees Windchill login form instead of ThingWorx login form.

3. User enters user name and password in the login form. Windchill authenticates the user.

4. After a successful login, Windchill server redirects request back to ThingWorx with the generated key.

5. The PTC Identity Provider key validation filter reads the key and validates it.

6. The validation filter passes the Windchill user name to the next filter as request attribute. In the single sign-on process, the next filter is the ThingWorx Authentication Filter.

   Authentication is complete.

7. The user display is updated with a ThingWorx window. The authenticator configuration determines the initial window for an authenticated user (as described in the next section).

For details on installing and configuring the filter, see [Missing cross reference text].

Design of Authenticator Extension

The purpose of the PTC Identity Provider Authenticator Extension is to authenticate and grant ThingWorx users access to the different parts of a ThingWorx application.

The authentication flow is described in the following process:

1. After the remote IdP completes the authentication and has passed off the Windchill user name, the existing ThingWorx Authentication Filter invokes the PTC Identity Provider Authenticator Extension.

2. PTC Identity Provider Authenticator Extension performs the following steps:

   a. Reads the user name from the request attribute.

   b. If the user does not exist in the ThingWorx system, the extension creates the user in ThingWorx.

   > 📋 **Note**
   >
   > The option to create new users is configurable. For more information, see Configuring the PTC Identity Provider Authenticator on page 34.

   c. Optional: If you have configured a home mashup, sets the home mashup for newly created users and gives the users read access to the mashup.

      If the user exists in ThingWorx, it does not set the user's home mashup.

3. After a successful authentication, the ThingWorx internal controller redirects the users to their home mashup or the default ThingWorx search page.

For configuration details and additional details about configuring users, see Configuring the PTC Identity Provider Authenticator on page 34.

C

Using PTC Data Query Extension when Developing Your Application

The PTC Data Query Extension facilitates rapid development of providing PTC data in ThingWorx mashups. Through a simple graphical interface, the extension allows you to automatically generate ThingWorx Data Shapes and services that can then be used for searching and navigating structures. Additionally, the extension provides a mechanism for specifying the desired attributes that you want returned from a Data Query (based on resource type).

The PTC Data Query Extension is installed when you install the PTC Windchill Extension ZIP file.

The PTC Data Query Extension has the following support limitations:

- You can use the PTC Data Query Extension when connecting to systems running Windchill 10.2 M020 and later; the extension is not supported when you are connecting to Windchill 10.1 M040 or 10.1 M050.

- Does not support the use of the extension with Internet Explorer 8 and Internet Explorer 9 browsers.

- Supports strictly binary link relationships; there is no support for foreign key references.
- Provides limited support of object types; supports the use of only Part, Document, and EPM Document.

The following sections explain how to configure the extension, launch the extension interface, and use the extension to create a Data Query.

Configuring the PTC Data Query Extension

You can use the PTC Data Query Extension from within the **ptc-windchill-demo-thing** by setting some configuration settings in **ptc-windchill-demo-thing** and creating a new data shape. Complete the following steps to configure the extension for use within **ptc-windchill-demo-thing**:

1. In the ThingWorx Composer navigation pane, under **MODELING**, select **Things**.

2. Edit **ptc-windchill-demo-thing**.

3. Under **ENTITY INFORMATION**, select **Configuration**.

4. In the **PTC Connection Settings** section, set values for the following fields:

   - **restPath**: Enter `/servlet/rest.`

   - **navigationParams**: Create a new data shape using the following procedure:

     a. Click the wand icon.

     b. In the data shape **Search Results** window, click **Data Shape** to create a new data shape.

     c. Under **General Information**, enter the name (for example, enter `com.ptc.dataquery.navigationParamsDataShape`) and a description of the shape.

     d. Under **Field Definitions**, click **Add**.
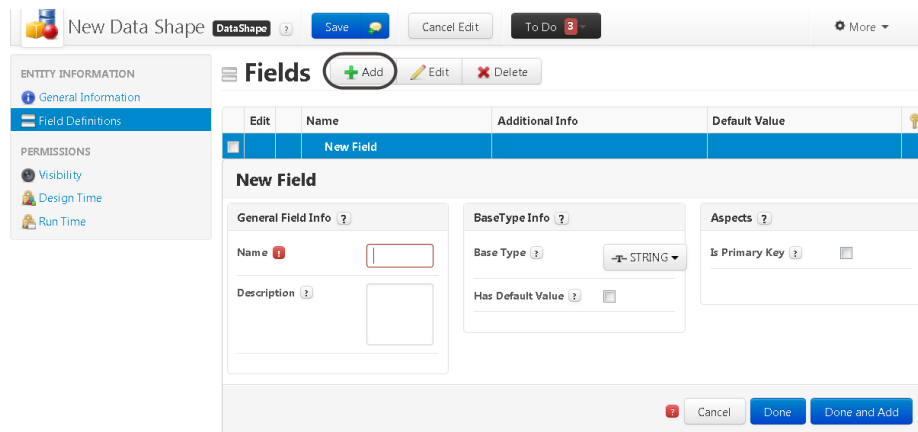
     i. In the **Name** field, enter `oid`.

     ii. Click **Done and Add**.

     iii. In the **Name** field, enter `returnRoot.`

     iv. In **Base Type**, select **Boolean**.

   v. Click **Done**

     e. Click **Save** to save the data shape.

The com.ptc.dataquery.navigationParamsDataShape shape has been created and is the value set in the **navigationParams** field for **ptc-windchill-demo-thing**.

5. Switch to the **ptc-windchill-demo-thing** tab and click **Save**.

Launching the Data Query Entity

After configuring the connection (as discussed in the previous section), launch the graphical interface of PTC Data Query.

1. In ThingWorx Composer, select **New Entity** and then select **Data Query**.
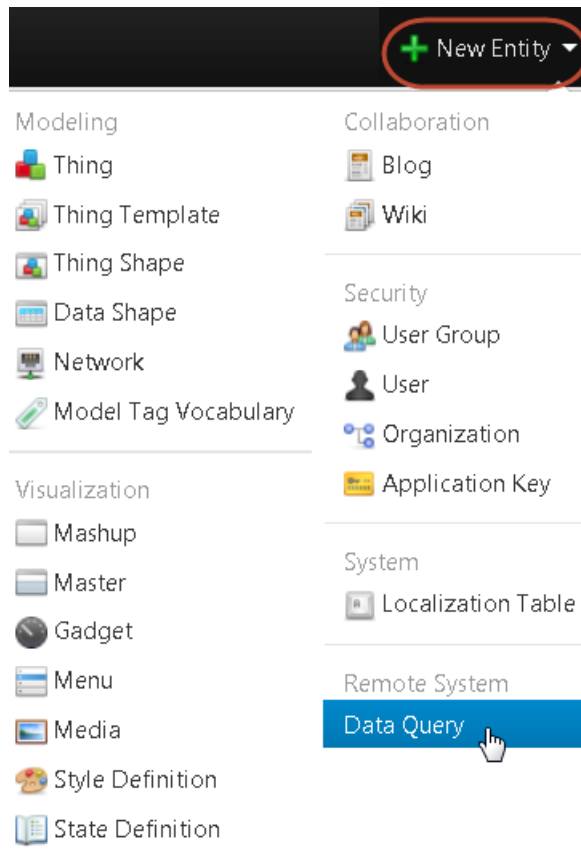
2. A **Select Link** window opens. Under **Available System Connections**, select **ptc-windchill-demo-thing** and then click **Create New**.
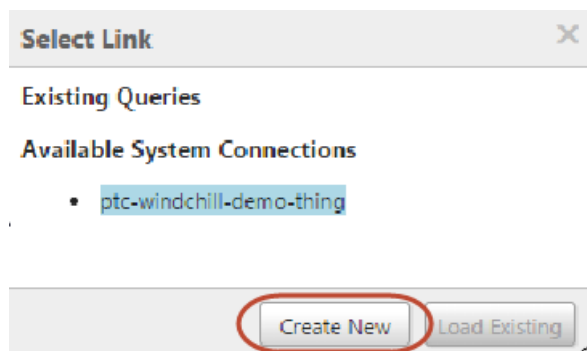
3. The following window opens. Follow the steps in the next section of this guide to use the PTC Data Query interface to create data queries.

# Using the Extension to Create New Data Queries

After you have launched the data query entity (as described in the previous section) and have selected to use an established thing entity (such as **ptc-windchill-demo-thing**), you can create a data query. At a high level, you need to accomplish the following activities:

1. Select data resources.
2. Link data sources.
3. Save the query.

The following sections describe how to accomplish these activities using **ptc-windchill-demo-thing**. While ideal for demo environments, you should create a new thing from the WindchillConnector thing template for production environments.

Selecting Data Resources

1. Select one or more of the following resources: **Document**, **EPM Document**, or **Part**:
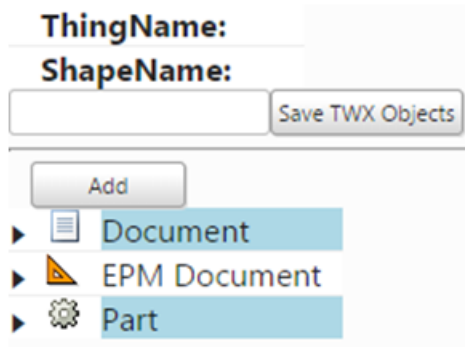

2. Click **Add** to add the resource to the work area. For example, the following screen shows both **Document** and **Part** in the linking pane of the window:
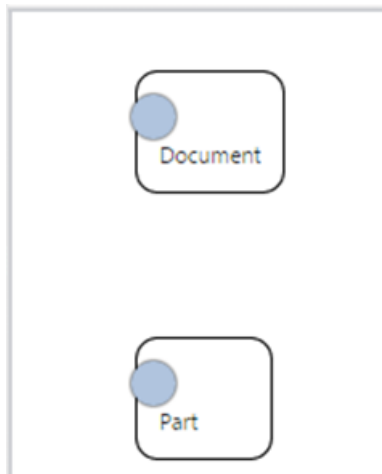

Linking Data Resources

1. In the linking pane, draw an arrow from the blue circle on one resource to the blue circle on another.
2. As soon as you click the blue circle of the second resource, a **Select Link** dialog opens showing a list of the relationships that can be selected between the two resources.

3. Select a relationship. For example, the following screen shows **WTPart Describe Link** selected:
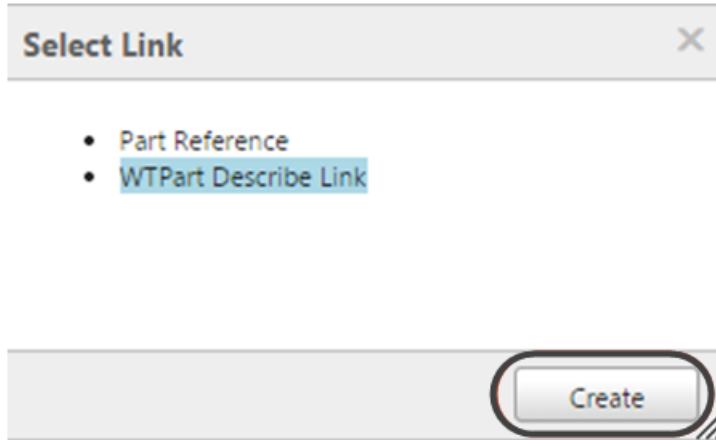
4. Click **Create** and the relationship appears as an arrow between the two resources:
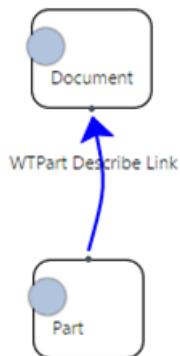
5. The right top panel is used to define a common alias for each resource

attribute and its relationship. To the right of the **Add Alias** button, enter an alias name in the input box and then click **Add Alias**.

6. The alias appears as a new row in the table. Select the mapped attribute names from a drop-down list under each column. For example, the following screen shows all of the available `Number` attributes:

Saving a Data Query

After defining all attribute aliases, save the query using the following steps:

1. Enter a name for the query in the field next to the **Save TWX Objects** button at the top of the panel.

2. Click **Save TWX Objects**.

A confirmation message displays the name of the data shape and thing entities that are created

Understanding the Saved Query ThingWorx Services

When you save a query, the newly created thing has two services:

• **getNaviationResults**

This service takes the fields of the data shape for **navigationParams** in the **ptc-windchill-demo-thing** configuration as the input parameters and returns the navigation result as an Infotable of the created data shape.

Using the previous example, **getNaviationResults** takes the **oid** of a part and returns the **Name** and **Number** attributes of its describing documents.

• **getSearchResults**

This service takes a **filter** parameter that uses `OData $filter` syntax to define the search condition and returns the search result as an Infotable of the created data shape.

Using the previous example, you could search any part or document using the search condition passed in the **filter** parameter (for example, `Number eq 12423`) and return the **Name** and **Number** attributes of the matching parts or documents.

Creating Example of a Mashup for a Part Number Search

The PTC Data Query Extension can be used to create a mashup that would allow users to search by a part number and then view attribute data of the returned parts. The following sections describe how to create this example mashup:

1. Create two ThingWorx Things in the PTC Data Query Extension:

    • The PartSearchExample Thing is used for searching.

    • The PartNavExample Thing is used for navigation.

2. Create a mashup that uses the Things created in step 1.

The mashup that is created allows a user to enter a partial part number and then select a part from the search results. Selecting a part displays the structure and reference documents of the part.

The resulting mashup can look similar to the following:

The following sections provide details about completing the steps to create a mashup similar to the previous example.

Creating the PTC Data Query Things

Before creating the mashup, create two Data Query Things using the PTC Data Query Extension interface:

• Create **PartSearchExample** as shown in the following:

Enter `PartSearchExample` as the name and click **Save TWX Objects**. Saving ThingWorx objects creates `com.ptc.dataquery.PartSearchExampleThing` and `com.ptc.dataquery.PartSearchExampleShape`.

The search object just created allows searching by name or number. However, this example searches only by number.

• Create **PartNavExample** as shown in the following:

Enter `PartNavExample` as the name and click **Save TWX Objects**. Saving creates a ThingWorx Thing and Data Shape.

Creating the Mashup Layout

Create an initial layout of the mashup with the following:

- A header

  In the header, place a text box, a button, and an expression widget.
- Two rows
  - In the first row, place a Grid.
  - in the second row, place a Tree Table.

The following screen shows the layout you want to create:
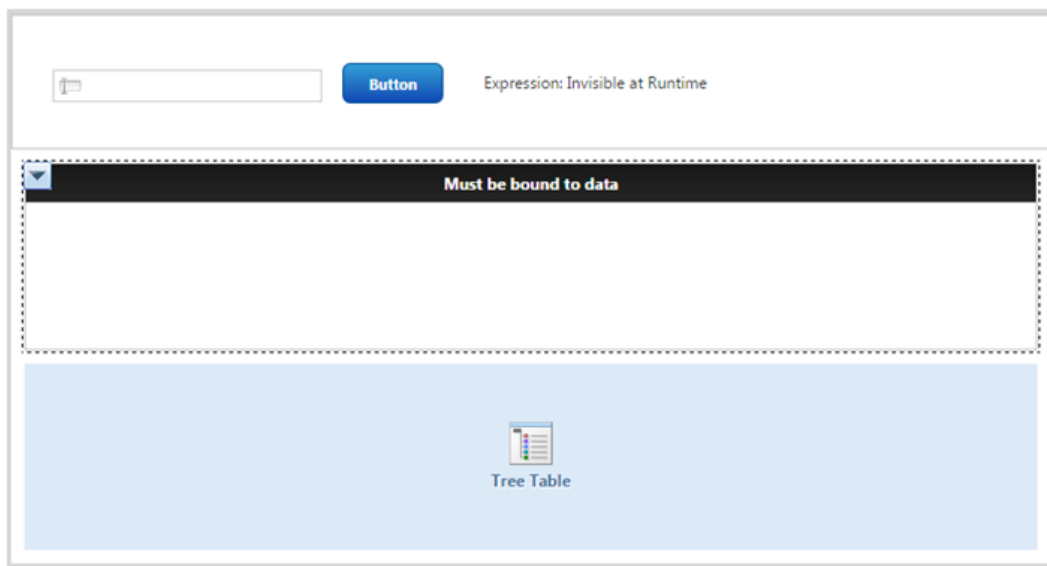
The following sections describe additional details about configuring the mashup layout content. When you finish implementing the details that are described, your mashup should look similar to the mashup shown at the beginning of this topic.

Configure Expression Widget

Complete the following steps to configure the expression widget:

1. Configure the expression to add a parameter called `numberSearch` of type **String**.

2. In the details panel of the expression:

   - Change the **OutputBaseType** to be of type **String**.
   - Set the `AutoEvaluate` property of the expression widget to `true`.
   - Set the expression to:

     ```
     encodeURI("substringof(number, '" + numberSearch +
     "')")
     ```

Add Services to Mashup

Add the following services:

- Locate `com.ptc.dataquery.PartSearchExampleThing` and add the **getSearchResults** service.
- Locate `com.ptc.dataquery.PartNavExampleThing` and add two instances of the **getNavigationResults** service.

Set Bindings in Mashup

Complete the following steps to create the bindings required to make the data appear correctly in the mashup:

1. Bind the **Text** of the text box to the **numberSearch** value of the expression widget.
2. Bind the **Output** of the expression widget to the filter parameter of the **getSearchResults** service just added.
3. Bind the Button's **Clicked** action to the **getSearchResults** service just added.
4. Bind **All Data** from the **getSearchResults** service to the Grid.
5. Bind the **objectId** of the **Selected Rows** in **getSearchResults** to the **oid** parameter of the first **getNavigationResults** service.
6. Bind the **SelectedRowsChanged** event of **getSearchResults** to execute the first **getNavigationResults** service.
7. Select the first **getNavigationResults** service and check the **returnRoot** property in the bottom sections of the services panel.
8. Bind the **All Data** of the first **getNavigationResults** service to the **Data** of the Tree Table.
9. Bind the **objectId** of the Selected Rows of the first **getNavigationResults** service to the **oid** parameter of the second **getNavigationResults** service.
10. Bind the **All Data** of the second **getNavigationResults** service to the **Children** of the Tree Table.
11. Bind the **GetChildren** event of the Tree Table to execute the second **getNavigationResults** service.

Configure Tree Table

Select the **Tree Table** in the mashup. In the details panel:

- Set the **IDField** to `rowId`
- Set the **ParentIDField** to `parentRowId`
- Set the **HasChildrenFlag** to `hasChildren`

Configure Columns of Gird and Tree Table

In the **Configure TreeTable Columns** and **Configure Grid Columns** of the respective widgets, hide all but the **name**, **number**, **quantity**, and **iconPath** columns. Then, select the **iconPath** columns and go to the **Column Renders** tab and set **Renderer** to **IMAGELINK**.

Save Mashup

Be sure to save your mashup as well as all entities that were changed during the configuration.