



# **PTC<sup>®</sup> Navigate Installation and Administration Guide**

**PTC<sup>®</sup> Navigate Framework Extension 1.0**

**PTC<sup>®</sup> Navigate View ALM App Extension 1.0**

**PTC<sup>®</sup> Navigate View PLM App Extension 1.1**

**PTC Windchill<sup>®</sup> Extension 1.2.1**

**March 2016**

---

# Copyright

**Copyright © 2016 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.**

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes. Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

**UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.**

PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

**Important Copyright, Trademark, Patent, and Licensing Information:** See the About Box, or copyright notice, of your PTC software.

---

## **UNITED STATES GOVERNMENT RIGHTS**

PTC software products and software documentation are “commercial items” as that term is defined at 48 C.F.R. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1 (a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

PTC Inc., 140 Kendrick Street, Needham, MA 02494 USA



# Contents

<b>PTC Navigate Framework Extension</b> .....	7
1 PTC Navigate Framework Extension Overview .....	8
2 PTC Navigate Release Updates .....	9
3 Installing PTC Navigate Framework Extension .....	10
4 Upgrading PTC Navigate .....	16
5 Known Limitations in PTC Navigate .....	20
<b>PTC Navigate View PLM App Extension</b> .....	23
6 PTC Navigate View PLM App Extension Product Overview .....	24
7 Installing PTC Navigate View PLM App Extension .....	27
8 Configuring PTC Navigate View PLM App Extension .....	33
9 Tailoring PTC Navigate View PLM App Extension .....	47
<b>PTC Navigate View ALM App Extension</b> .....	73
10 PTC Navigate View ALM App Extension Product Overview .....	74
11 Installing PTC Navigate View ALM App Extension .....	76
12 Configuring PTC Navigate View ALM App Extension .....	79
13 Tailoring PTC Navigate View ALM App Extension .....	83
14 Known Limitations PTC Navigate View ALM App Extension .....	86
<b>Appendices</b> .....	89
Appendix A. Developing a Customized Application .....	91
Appendix B. Example Configuration Using SSL for Secure Communications .....	105
Appendix C. Architecture of Single Sign-On .....	109

---

## About This Guide

This guide documents the installation and administration of the components required for the PTC Navigate.

### Comments

PTC welcomes your suggestions and comments on its documentation. To submit your feedback, you can send an email to [documentation@ptc.com](mailto:documentation@ptc.com). To help us more quickly address your concern, include the name of the PTC product and its release number as well as the book title.

## Technical Support

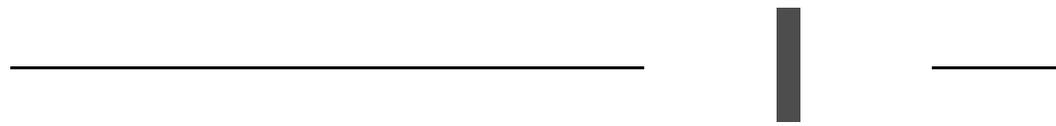
Contact PTC Technical Support through the PTC website, or by phone, email, or fax if you encounter problems using this product or the product documentation. The PTC eSupport portal provides the resources and tools to support your PTC Windchill implementation:

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

For complete 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.



# PTC Navigate Framework Extension

The PTC Navigate Framework Extension provides the platform for the apps you use in PTC Navigate. Install the framework and then proceed to the section for the PTC product apps of your choice.

---

# 1

## PTC Navigate Framework Extension Overview

PTC Navigate provides viewing apps that allow users to quickly and easily access specific information stored in PTC products. The apps that you can view on the PTC Navigate landing page are based on the product extensions you have imported. The PTC Navigate Framework Extension provides a platform for the app extensions. For information on the apps available, see [PTC Navigate View PLM App Extension Product Overview on page 24](#) and [PTC Navigate View ALM App Extension Product Overview on page 74](#).

# 2

## PTC Navigate Release Updates

The following list highlights updates to PTC Navigate.

### Release 1.0 to 1.1

- PTC Navigate includes the PTC Navigate View ALM App Extension, which contains the View Requirement app.
- PTC Navigate delivers separate extensions, such that the PTC Navigate Framework Extension is a separate extension from the PTC Navigate View PLM App Extension and PTC Navigate View ALM App Extension.
- Administrators can view which extensions are deployed on the ThingWorx server from the training drop-down list  on the PTC Navigate landing page.
- The following bugs are fixed:
  - Drawing cannot be found when there are multiple parts having the same number and different views
  - Windchill views were not being properly displayed in **Part Collection Tailoring** page.

---

# 3

## Installing PTC Navigate Framework Extension

This chapter describes the installation steps for PTC Navigate Framework Extension. Once the PTC Navigate Framework Extension is installed, proceed to the installation for the app extension your site will use.

### PTC Navigate Framework Extension Product Support

The following sections detail the product support for PTC Navigate Framework Extension. For information on the app-related product requirements, see [PTC Navigate View PLM App Extension Product Support on page 27](#) and [PTC Navigate View ALM App Extension Product Support on page 76](#).

#### Server Platform Support

- ThingWorx 6.5 only  
Additional information can be found in the [ThingWorx 6.5 system requirements](#).
- 6.5 PostgreSQL or 6.5 Neo4j
- ThingWorx Converge 6.5.0.01
- Tomcat 8

If another instance of ThingWorx has been deployed on the system prior to this installation, you must delete the following folders that were created in the base directory where Tomcat was installed:

- ThingworxBackupStorage

- 
- ThingworxStorage
  - Window Server 2012
  - Red Hat Enterprise Linux (RHEL) 7 Update 1

### Client Platform Support

- Java 8
- Browser support
  - Internet Explorer 10
  - Firefox 35 and later
  - Chrome 44
  - Safari 6.1.6 and later

---

#### Note

- 3-D Visualization is supported on Windows clients with Firefox or Internet Explorer only.
- 

## Installation Overview

The following topic provides an overview of the process to download and install the required components for PTC Navigate. All components, other than ThingWorx 6.5, can be downloaded from the PTC Smart Connected Applications section of the [PTC Software Downloads](#) page.

### 1. ThingWorx

To install and configure, see [Installing ThingWorx 6.5](#).

### 2. PostgreSQL

PostgreSQL is included with the purchase of PTC Navigate and can be used instead of Neo4j. The [Installing ThingWorx 6.5](#) guide describes the differences between Neo4j and PostgreSQL.

To install and configure PostgreSQL, see [Getting Started with PostgreSQL and ThingWorx 6.5](#).

### 3. ThingWorx Converge

Importing ThingWorx Converge is described in [Importing ThingWorx Converge Files on page 12](#).

### 4. PTC Navigate Framework Extension

---

Importing PTC Navigate Framework Extension is described in [Importing PTC Navigate Framework Extension Files](#) on page 15.

5. Product app extensions

- PTC Navigate View PLM App Extension

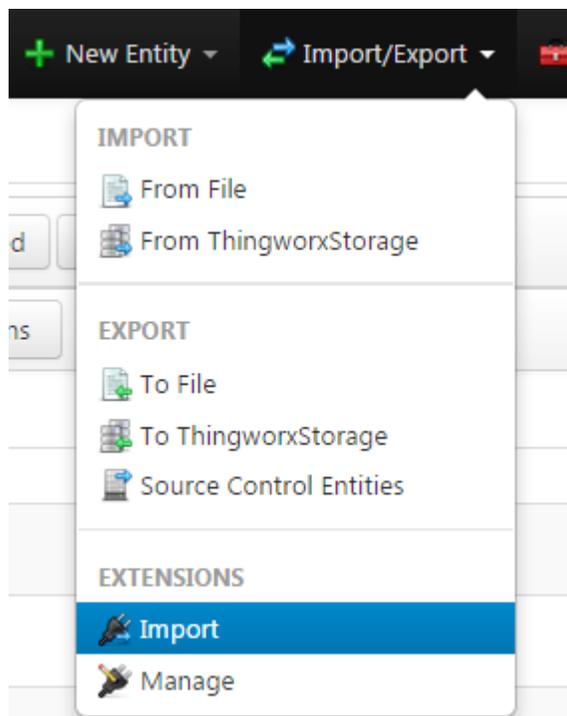
Import the app extension files as well as patches specific to the version of Windchill you have installed. For more information, see [Installing PTC Navigate View PLM App Extension](#) on page 27.

- PTC Navigate View ALM App Extension

For more information, see [Installing PTC Navigate View ALM App Extension](#) on page 76.

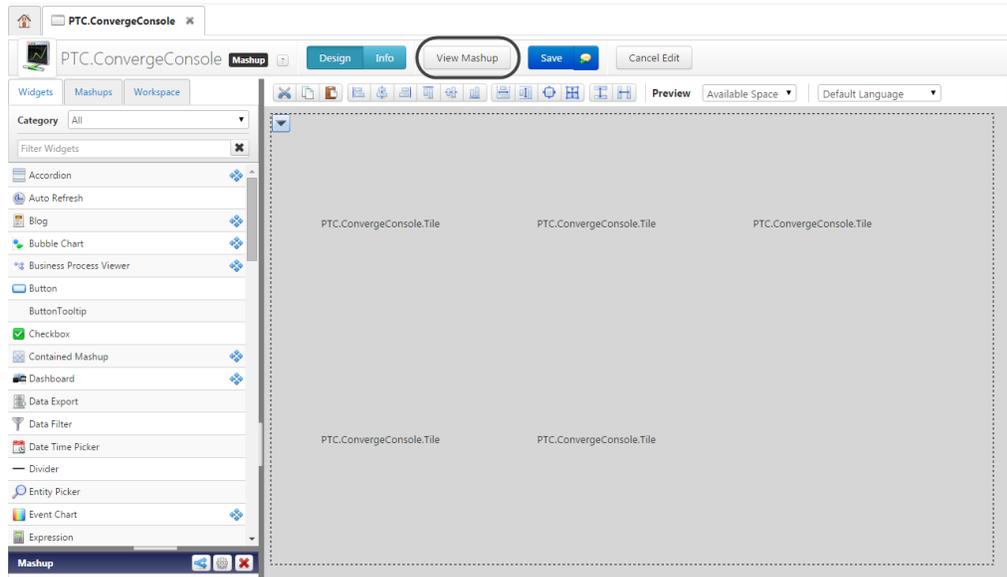
## Importing ThingWorx Converge Files

1. From the ThingWorx Composer, navigate to **Import/Export** ► **Import**.

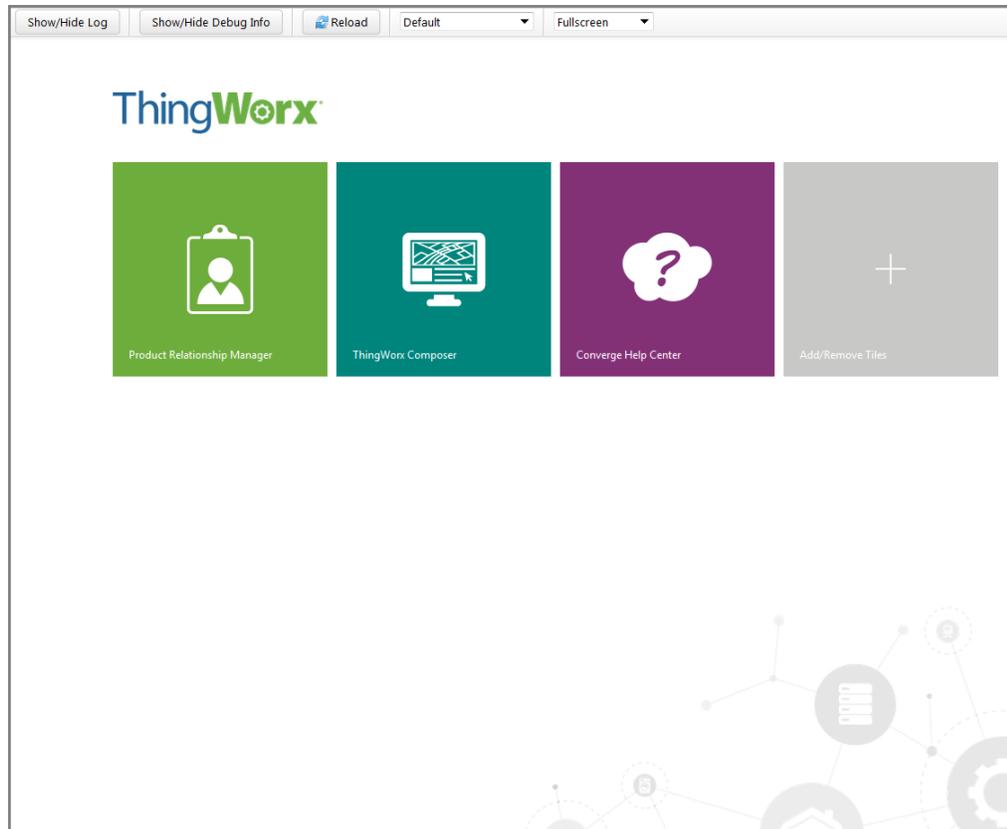


2. The **Import Extensions** window appears.
3. Navigate to `TWX_Converge_Core_ExtensionPackage-{version}-extension-bundle.zip`, and click **Import**.
4. Click **Close**. When prompted to refresh your browser, click **Yes**.

- Verify that the extension has been successfully imported by searching for the **PTC.ConvergeConsole** mashup type. Select the mashup by clicking the name or the view icon. From the Mashup Editor, click **View Mashup**.



- The mashup opens:



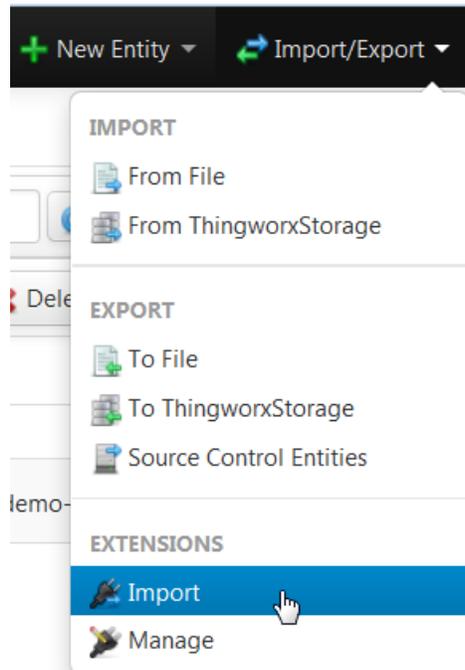


---

## Importing PTC Navigate Framework Extension Files

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

1. Download and unzip the contents of the following file: PTC-Navigate-Framework-`{version}`.zip
2. From ThingWorx Composer, navigate to **Import/Export** ► **Import**:



3. In the **Import Extensions** window, navigate to `ptc-navigate-framework-extension-{version}.zip`.
4. Click **Import** and refresh the window if prompted.
5. Again select **Import/Export** ► **Import**.
6. Navigate to `ptc-identity-provider-authenticator-extension-{version}.zip` and click **Import**.

To view the extensions after the import is complete, select **Import/Export** ► **Manage**.

7. Import files for the app extension your site will use.
  - [Importing PTC Navigate View PLM App Extension Files on page 30](#)
  - [Importing PTC Navigate View ALM App Extension Files on page 77](#)

---

# 4

## Upgrading PTC Navigate

The following process describes the upgrade from release 1.0 to 1.1 of PTC Navigate.

---

### Note

- These upgrade steps are for PTC Navigate only. If you are upgrading ThingWorx or ThingWorx Converge, those prerequisite products should be upgraded in the correct sequence. For information on the installation order, see [Installation Overview on page 11](#).
- If your site is on Windchill 10.2 M020, before upgrading PTC Navigate, apply the latest CPS for Windchill 10.2 M020. If your site is already on the latest CPS for Windchill 10.2 M020, re-apply the CPS before upgrading PTC Navigate.

---

### 1. Backup Configuration Data

Before removing a previous release of PTC Navigate, backup your configuration using ThingWorx import and export functionality. PTC recommends exporting all entities in order to back up all data. Use the following procedure:

1. At the top of the ThingWorx Composer screen, select **Import/Export ► Export ► Source Control Entities**.

The **Export Source Control Entities** window opens.

2. In the **Collections** field, select **All**. Other fields can retain their default values.
3. Click **Export**.

---

The system repository is in the following directory: ThingworxStorage\repository\SystemRepository.

4. If your site added Windchill lifecycle states in addition to the states provided by default in PTC Navigate, complete the following steps to migrate this data:
  - a. Select **Import/Export ► Export ► To File**.

The **Export to File** window appears.

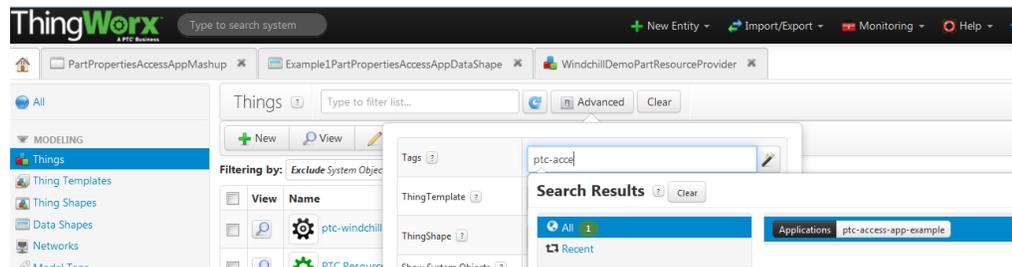
- b. Click the **Data** tab, and in the **Collections** field, select **DataTables**.
- c. Click **Export**.

A new directory is created in the system repository, for example, ThingworxStorage\repository\SystemRepository\20160212163521\DataTables\PTC.AccessApp.LifecycleStateDataTable.

## 2. Remove PTC Navigate

Use the following procedure to remove PTC Navigate files from ThingWorx Composer in preparation for an upgrade.

1. From ThingWorx Composer, use **Advanced** search to search by tag for applications with `ptc-access-app-example`.



2. Delete all things in the search results.
3. Also delete the following thing: **SAPDemoPartResourceProvider**.
4. At the top of the ThingWorx Composer screen, select **Import/Export ► Extensions ► Manage**.
5. Delete the following extensions:
  - MockProviders\_ExtensionPackage
  - PTC-Navigate-View-PLM-App-extension
  - WindchillOntology\_ExtensionPackage
  - PLMOntology\_ExtensionPackage
6. Restart the Tomcat server.

### 3. Install PTC Navigate

1. Install the extensions for PTC Navigate in the required order:
  - a. [Importing ThingWorx Converge Files on page 12](#)
  - b. [Importing PTC Navigate Framework Extension Files on page 15](#)
  - c. [Importing PTC Navigate View PLM App Extension Files on page 30](#)  
 When upgrading, it is not necessary to re-import `ptc-windchill-sample-mashup-{version}.xml`.
  - d. [Importing PTC Navigate View ALM App Extension Files on page 77](#)
2. Restart the Tomcat server.

The PTC Navigate 1.1 landing page will be available with all default tailoring options.

### 4. Restore Configuration Data

After installing the extensions for the new release, restore the configuration data you backed up previously.

1. At the top of the ThingWorx Composer screen, select **Import/Export ► Import ► From File**.
2. Using the table below, import the XML files with the corresponding app configurations.

Note that upon export, the system repository was in the following location:  
`ThingworxStorage\repository\SystemRepository`.

part and part app collection	Thing\Example1DesignFilesAccessAppConfigThing.xml Thing\Example1DrawingAccessAppConfigThing.xml Thing\Example1InterrogateCadModelAccessAppConfigThing.xml Thing\Example1PartPropertiesAccessAppConfigThing.xml Thing\Example1PartsListAccessAppConfigThing.xml Thing\Example1PartStructureAccessAppConfigThing.xml
view document app configuration	Thing\Example1DocumentAccessAppConfigThing.xml
recently viewed items	Thing\RecentlyViewedItemsThing.xml

- 
3. If your site is migrating Windchill lifecycle states in addition to the states provided by default in PTC Navigate, complete the following steps:
    - a. Select **Import/Export ► Import ► From File**.  
The **Import from File** window opens.
    - b. Select the **Data** button and the **Single File** tab.
    - c. Use the **Browse** button to navigate to the lifecycle state file exported previously, for example, `DataTables\  
PTC.AccessApp.LifecycleStateDataTable\data-0.twx`.
    - d. Click **Import**.
  4. Restart the Tomcat server.  
Tailoring options and recently searched items are carried forward to the upgraded version of PTC Navigate.

## 5. Re-apply User Permissions

After an upgrade, user permissions will need to be set again. For more information, see [Modify ThingWorx Run Time Permissions on page 42](#).

---

# 5

## Known Limitations in PTC Navigate

The following issues describe circumstances that may exist in PTC Navigate Framework Extension or PTC Navigate View PLM App Extension. For information on PTC Navigate View ALM App Extension, see [Known Limitations PTC Navigate View ALM App Extension on page 86](#).

### Form-Based Authentication

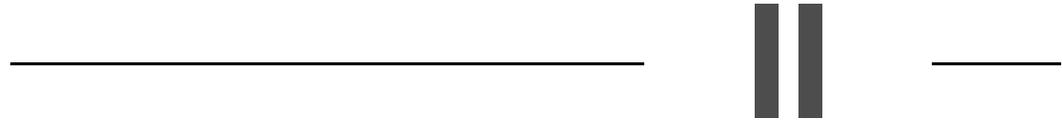
- Issue: PTC Navigate is not working when the server is set up to use form-based authentication.
- Workaround:
  - For Windchill servers, use Tomcat authentication.
    1. Navigate to `$WT_HOME/tomcat/configAuth`.
    2. Run the following command: `ant -f authConfig.xml enable`
  - When using form-based authentication with another enterprise solution, use the following steps:
    1. Navigate to `<WT_HOME>/codebase/WEB-INF/web.xml`.
    2. Move this line of code: `<url-pattern>/sslClientAuth/*</url-pattern>`  
from this security constraint `<web-resource-name>Authenticated resources</web-resource-name>`  
to this security constraint `<web-resource-name>Anonymously accessible resources</web-resource-name>`
- Resolution: This issue is resolved in Windchill 11.0 M010.

---

## Common Tailoring Page Selections for Saved Filter and Latest Filter Types

- Issue: On the **Common Tailoring** page, selections for **Saved Filter** or **Latest** filter types do not appear after an administrator's session expires; however, the selections remain persisted in the app.
- Workaround: If an administrator returns to the **Common Tailoring** page to make additional changes, re-select the desired options for **Saved Filter** or **Latest** filter types.
- Resolution: This issue will be resolved in a future release of PTC Navigate View PLM App Extension.





# **PTC Navigate View PLM App Extension**

# 6

## PTC Navigate View PLM App Extension Product Overview

PTC Navigate View PLM App Extension provides viewing apps that allow users to quickly and easily access specific information stored in Windchill products. The following apps are available with this release.



Each of these view apps provides a search field in which you enter criteria for the object you want to view. The available search fields and their function can be modified through app tailoring. For more information, see [Basic App Tailoring on page 47](#).

This landing page is a ThingWorx mashup available at the following URL:  
`http://<host>:<port>/Thingworx/Runtime/index.html#master=PTC.AccessApp.Master&mashup=LandingPageAccessAppMashup`

---

** Best Practice**

- PTC Navigate apps were designed for a screen resolution of 1280x1024. Results on other resolutions may vary.
- App searches may need extra time when calling Windchill data. For more information, see [Editing Thing Configuration on page 45](#).

---

**App Collections**

Apps are grouped into related collections. For this release, part-related apps are in a collection, separate from the View Document app. The app collections provide the following benefits:

- Users can navigate to other apps in the collection using the icons available from the details pages. For example, when viewing part properties, you can select the view design files icon and switch to that app.

PTC® Navigate View Part Properties

← Back To Search

PART DETAILS



Number	000000097
Name	bike-ptc
Version	1.2 (Design)
State	Design



ADDITIONAL ATTRIBUTES

Last Modified 2015-10-01 11:12:22

- Administrators can tailor the app functionality as a collection. For example, when deciding on which attribute the apps search (name, number, or both), you set the option once rather than tailor each app individually. For more information, see [Basic App Tailoring on page 47](#).

### Session Timeout

If you see a message similar to the following image, your session has timed out. Click **Dismiss** and refresh your browser to return to your current session.

Status Message Dismiss

 Could not load "DrawingAccessAppMashup". Reason: 0 -

# 7

## Installing PTC Navigate View PLM App Extension

This chapter describes the installation steps for PTC Navigate View PLM App Extension.

### PTC Navigate View PLM App Extension Product Support

The following information details the product support for PTC Navigate View PLM App Extension.

- Windchill at one of the following release levels:
  - 11.0
  - 10.2 M030 CPS 01 or later
  - 10.2 M020 CPS 06 or later
  - 10.1 M040 or M050

---

#### Note

Windchill system requirements information for your specific Windchill release is available on the [Reference Documents](#) page.

---

- Windchill patch that corresponds with the release you have installed and is included with PTC Navigate. For more information, see [Installing PTC Windchill Patches on page 31](#).

- 
- PTC Windchill Extension 1.2.1, included with PTC Navigate View PLM App 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 33](#). The required Windchill security fixes are available from PTC:

- For Windchill 10.1 M040 and 10.1 M050, upgrade PTC HTTP Server using the latest 10.1 Early Release Download that is available from PTC Technical Support:

[http://support.ptc.com/appserver/wcms/standards/freefull.jsp?im\\_dbkey=73856&icg\\_dbkey=893](http://support.ptc.com/appserver/wcms/standards/freefull.jsp?im_dbkey=73856&icg_dbkey=893)

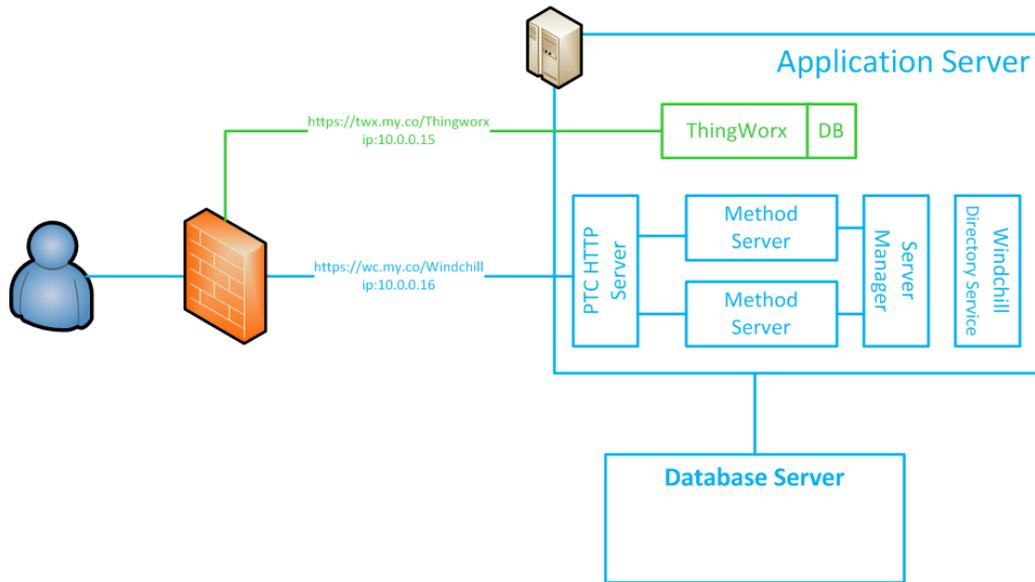
For upgrade instructions, see the `Installing_Apache.doc` file that is included in the ZIP that you download.

- 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 105](#).

- PTC Creo View 3.0 M031 for the **View Part Structure** app

The following diagram depicts a ThingWorx deployment with Windchill.

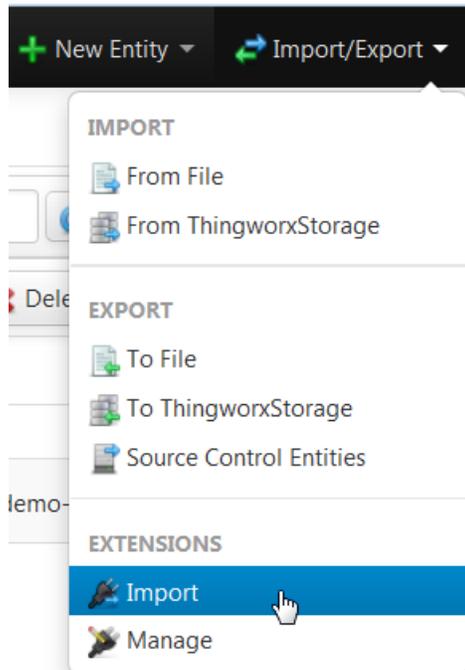


---

## Importing PTC Navigate View PLM App Extension Files

To install the extensions included with the PTC Navigate View PLM App Extension, complete the following steps:

1. Download and unzip the content of the following file: PTC-Navigate-View-PLM-Apps-{version}.zip.
2. From ThingWorx Composer, navigate to **Import/Export** ► **Import**:



3. In the **Import Extensions** window, navigate to PTC-Navigate-View-PLM-App-Extension-Bundle-{version}.zip.
4. Click **Import** and refresh the window if prompted.
5. Select **Import/Export** ► **From File**.
6. In the **Import from File** window, browse to ptc-sample-mashup-{version}.xml, and click **Import**.
7. Again select **Import/Export** ► **From File**.
8. Browse to PTCNavigate\_Example-{version}-imports.xml and click **Import**.

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
core-ui-ext	PTC Core UI Library
ptc-identity-provider-authenticator-extension	PTC Identity Provider Authenticator Extension
ptc-creo-view-extension	PTC Creo View Widget
linkTooltip_ExtensionPackage	Tooltips on Link
TWX_Converge_Styles_ExtensionPackage	ThingWorx Converge Common Style Definitions
MockProviders_ExtensionPackage	
WindchillOntology_ExtensionPackage	
GoogleWidgets_ExtensionPackage	Google Widgets
navigationTooltip_ExtensionPackage	Navigation Widget with Tooltip
PLMOntology_ExtensionPackage	
dataquery-ext	PTC Data Query
TWXClientCoreLibs_Extension	AngularJS Thingworx Connector
TWX_Converge_Core_ExtensionPackage	ThingWorx Converge Core

## 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.

# 8

## Configuring PTC Navigate View PLM App Extension

This chapter contains configuration information for ThingWorx and Windchill systems. Systems using the PTC Navigate View ALM App Extension may also choose to follow these configuration steps to use the single sign-on authentication solutions because Windchill is used as the identity provider for ThingWorx.

### Recommended Configurations

Not all configuration steps are necessary in every product installation. 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 sections with detailed steps. Choose a recommended configuration and follow the links to perform the configuration steps.

---

## 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 that is based on SSL public/private keys. Follow the steps in [Using SSL for Secure Communication on page 34](#).
- A single sign-on prompt for users. The single sign-on uses Windchill credentials. Follow the steps in [Configuring ThingWorx Single Sign-On when Connecting to Windchill on page 38](#).

## 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 37](#).
- Single sign-on is not required; however, relevant run-time permissions need to be set for every user on the system. Follow the steps in [Modify ThingWorx Run Time Permissions on page 42](#).

## 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 37](#).
- A single sign-on prompt for users. The single sign-on uses Windchill credentials. Follow the steps in [Configuring ThingWorx Single Sign-On when Connecting to Windchill on page 38](#).

# 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 [Recommended Configurations on page 33](#), PTC recommends using SSL when working in a production environment. For development and demo environments, see [Using Trusted Host on page 37](#). 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 [Example Configuration Using SSL for Secure Communications on page 105](#). This guide also assumes that both ThingWorx and Windchill are configured to use SSL for their standard communications.

---

 **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	<input type="text" value="JKS"/>
SSLKeyStorePath	<input type="text" value="/usr/local/tomcat/twx.jks"/>
SSLTrustStorePath	<input type="text" value="/usr/local/tomcat/trustedstore_ptc.jks"/>
SSLKeyStorePassword	<input type="button" value="Change Password"/>
SSLTrustStorePassword	<input type="button" value="Change Password"/>
SSLTrustStoreType	<input type="text" value="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.
  - 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.
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 to export the certificate information to Windchill.
  - a. Edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.
  - b. Find the line with `SSLOptions`.
  - c. Add a new line `SSLOptions +ExportCertData`.
  - d. Save changes.
4. 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</filter-name>`.
    - 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 34](#).

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 109](#).

Single sign-on requires that the same users exist in both Windchill and ThingWorx. The 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.

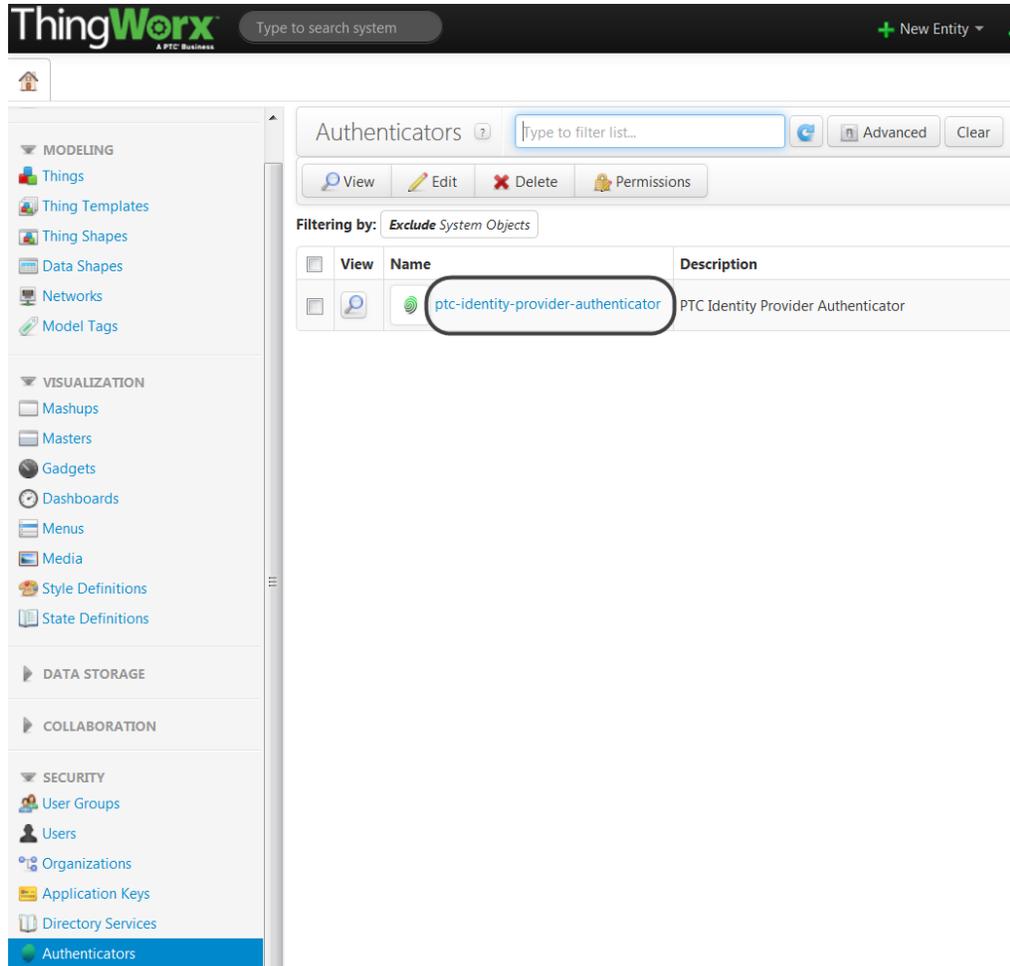
Use the steps in the following sections to configure single sign-on using Windchill as the identity provider:

1. [Configuring the PTC Identity Provider Authenticator on page 39](#)
2. [Add Servlet Filters to Tomcat Configuration on page 41](#)
3. [Modify ThingWorx Run Time Permissions on page 42](#)
4. [Verifying Single Sign On on page 45](#)

## 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.

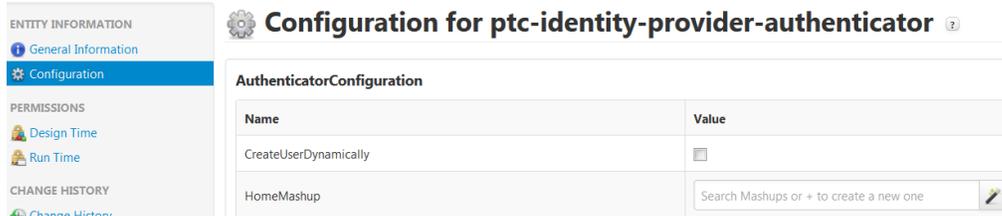


The **General Information** page opens.

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:



The screenshot shows a web interface for configuring an authenticator. On the left is a navigation pane with sections: ENTITY INFORMATION (General Information, Configuration, PERMISSIONS (Design Time, Run Time), CHANGE HISTORY (Change History)). The main area is titled 'Configuration for ptc-identity-provider-authenticator'. Below this is a table 'AuthenticatorConfiguration' with columns 'Name' and 'Value'. The table contains two rows: 'CreateUserDynamically' with a checkbox in the Value column, and 'HomeMashup' with a search input field containing the text 'Search Mashups or + to create a new one'.

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**.

---

 **Note**

If ThingWorx Tomcat is restarted, the **HomeMashup** selection is not preserved.

---

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 38.

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.
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 the following code blocks to the ThingWorx `web.xml` file before the `AuthenticationFilter` block.

---

### 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.

```

<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]/
      [windchill-web-app]/login/validateIdKey.jsp</param-value>
  </init-param>
</filter>
<filter-mapping>
  <filter-name>IdentityProviderKeyValidationFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>

```

## 5. Restart Tomcat.

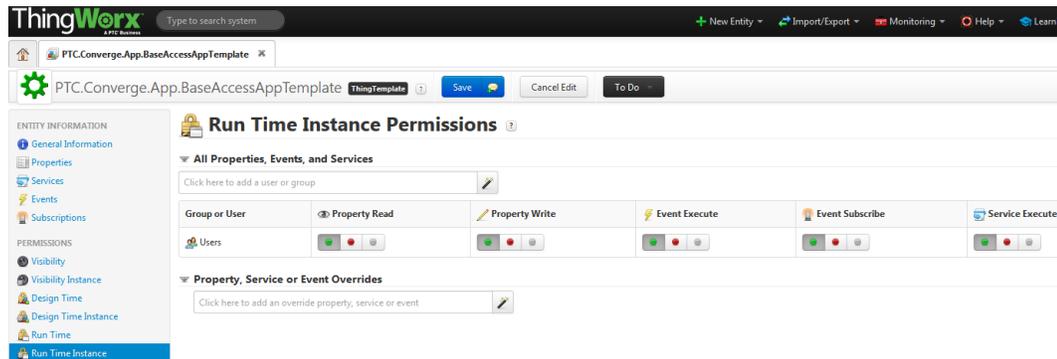
# Modify ThingWorx Run Time Permissions

## Permissions for Administrators

To be a PTC Navigate administrator, you 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 Role-based Configurations

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 as shown in this screen. Additional details are provided in the procedure below.



The following procedure describes the permissions to set in order to make certain apps visible to a given group.

### Note

SSO must be configured, or for non-production configurations, the user must exist in Windchill and have access to the products being searched from the PTC Navigate app.

1. Create user group (treated as user role) and add specific users to this group.
2. Under **Run Time** permission, give **Property Read** and **Service Execute** access to each thing that corresponds to the app for which you want the user to have access.

App	Configuration Things
Design Files	Example1DesignFilesAccessAppConfigThing
Drawing	Example1DrawingAccessAppConfigThing
View and Measure in 3D	Example1InterrogateCadModelAccessAppConfigThing
Part Properties	Example1PartPropertiesAccessAppConfigThing
Parts List	Example1PartsListAccessAppConfigThing

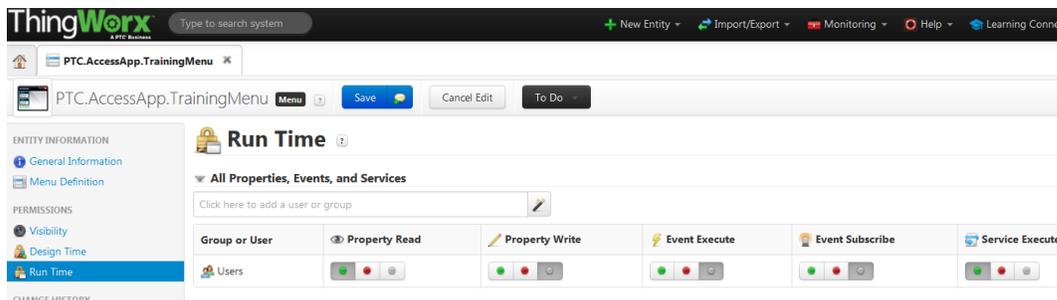
App	Configuration Things
Parts Structure	Example1PartStructureAccessAppConfigThing
Document	Example1DocumentAccessAppConfigThing

- Set all needed permissions for entities as described in the table below. The descendent things and thing templates related to PTC Navigate inherit the permission setting.

Entity	Permission	Permission Type
AppCollectionsConsolidator	Run Time	Property Read Service Execute
MasterAccessAppThing	Run Time	Property Read Service Execute
PTC.Converge.App. BaseAccessAppTemplate	Run Time Instance	Property Read Service Execute
PTC.Converge.App. RecentlyViewedItemsTemplate	Run Time Instance	Property Read Property Write Service Execute
PTC.Resource. ResourceProviderTemplate	Run Time Instance	Property Read Service Execute

## Permissions for Training Menu

The training menu  available in the upper right corner of the PTC Navigate screen is managed by permissions on the following entity: **PTC.AccessApp.TrainingMenu**. By default, the ThingWorx Users group has run time Read and Execute permissions for this entity. If the permissions are removed, the training menu does not display for non-administrative users.



The screenshot shows the PTC Navigate interface with the following elements:

- Header: ThingWorx logo, search bar, and navigation links (New Entity, Import/Export, Monitoring, Help, Learning Center).
- Entity Name: PTC.AccessApp.TrainingMenu
- Entity Type: Run Time
- Permissions Table:

Group or User	Property Read	Property Write	Event Execute	Event Subscribe	Service Execute
Users	<input checked="" type="checkbox"/>				

---

## 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.

## Editing Thing Configuration

Use the following steps to edit the configuration of **ptc-windchill-demo-thing** for PTC Navigate.

1. In ThingWorx Composer, navigate to **ptc-windchill-demo-thing**.
2. Under **ENTITY INFORMATION**, select **Configuration**.
3. Enter information in the following fields:
  - **baseURL**: The URL to the Windchill system. If using an https URL, Windchill must be configured for SSL as described in step 10 of the following sections:
    - [Using SSL for Secure Communication on page 34](#)
    - [Example Configuration Using SSL for Secure Communications on page 105](#)

- 
- **ServiceTimeout:** Time, in milliseconds, that calls to Windchill wait for a response before they timeout. If your app search does not return results, increasing the time in this field could resolve the issue.
  - **restPath:** Do not edit this field.

The remainder of the fields are not used.

4. Click **Save**.

# 9

## Tailoring PTC Navigate View PLM App Extension

The chapter provides information on modifying the display of the PTC Navigate View PLM App Extension.

### Basic App Tailoring

PTC Navigate provides administrators basic tailoring capabilities related to the app display and function. When you access the landing page as an administrator, edit icons appear for the apps. This page is available at the following URL:

```
http://<host>:<port>/Thingworx/Runtime/index.html#master=PTC.AccessApp.Master& mashup=LandingPageAccessAppMashup
```



---

## Note

To be a PTC Navigate administrator, you need to be added to the ThingWorx Administrators group. For more information, see [Modify ThingWorx Run Time Permissions on page 42](#).

---

- Click the edit icon  on individual apps to tailor options specific to the app.
- Click the edit collection icon  to access tailoring capabilities common to the set of related apps. For example, the part-related apps are in a collection, allowing you to tailor a number of related options at once.

## Collection Tailoring

The following table describes the options available from the **Part Collection**

**Tailoring** page, available when you click the edit collection icon .

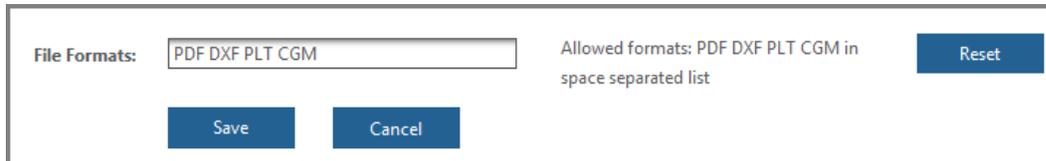
Field	Description
<b>Search Results Display Limit</b>	Maximum number of objects that display in the <b>Search Results</b> for the app.
<b>Recent Searches Display Limit</b>	Maximum number of objects that display in the <b>Recent Searches</b> for the app.
<b>Recent Searches Data Shape</b>	Data shape used in ThingWorx Composer that controls the display of the recent searches in the app. For more information, see <a href="#">Data Shape and Mashup Information on page 56</a> .
<b>Use CAD Documents as Parts</b>	Search for CAD documents instead of parts.
<b>Allow Partial Matches in Search</b>	Enable search results to return more than exact matches.
<b>Search on Attribute</b>	Select whether to search on the object number, name, or both.
<b>Enable Search Filter</b>	Select the checkbox to enable the <b>Select Filter Type</b> field from which you can select a filter to be available when users search in the apps.

Field	Description
<b>Select Filter Type</b>	<p>Select a filter to be available when users search from the apps:</p> <ul style="list-style-type: none"> <li>• <b>Saved Filter:</b> The saved filter displays in a drop-down list in the app search. For more information on saved filters, see the description for <b>Default Saved Filter</b> below.</li> <li>• <b>Saved Search:</b> The saved searches display in a drop-down list in the app search. Select saved searches available in the Windchill site context. For additional information, see the last section of this topic “Best Practices for Saved Searches.”</li> <li>• <b>Latest:</b> The lifecycle state and view by which users can filter their search. You can select multiple options from the list. Note that the search results return the latest iterations of all object revisions. To return only the latest iteration of the latest revision, create a save search as described in the last section of this topic “Best Practices for Saved Searches.”</li> <li>• <b>Version:</b> A <b>Version</b> field displays in the app search, enabling users to filter their search by an object version, for example A.1. Users must also enter information in the search field.</li> </ul> <p> <b>Note</b> This field is available when the <b>Enable Search Filter</b> checkbox is selected.</p>
<b>Default Saved Filter</b>	<p>Enter the exact name of the filter in Windchill. The app search displays object versions based on the filter.</p> <p>You create a filter from an object information page in Windchill. From the <b>Structure</b> tab, edit the filter to your desired settings. After saving the filter, copy and paste the name in this PTC Navigate field.</p> <p> <b>Note</b> This field is available when the <b>Enable Search Filter</b> checkbox is cleared.</p>

---

## App-specific Tailoring: File Formats for View Drawing, View Design Files, View Parts List

The tailoring pages for View Drawing, View Design Files, View Parts List include a **File Format** field. Select the edit icon  next to the app to launch the app-specific tailoring page.



The screenshot shows a tailoring window with a text input field labeled "File Formats:" containing the text "PDF DXF PLT CGM". To the right of the input field, there is a label "Allowed formats: PDF DXF PLT CGM in space separated list" and a "Reset" button. Below the input field are two buttons: "Save" and "Cancel".

- **View Drawing:** This app automatically downloads the drawing that matches the number entered. The file format downloaded is based on the order you specify in the **File Formats** field on the **View Drawing Tailoring** window as shown above. For example, if the order is PDF DXF CGM, and the number matches both a PDF and DXF file, the PDF file is returned because it is listed first in the **File Formats** field.
- **View Design Files:** This app displays the design files associated with the default representation of the part or CAD document number you enter. Enter data in the **File Formats** field to filter the search by only those formats.

If the **File Formats** field is blank, all default representations are returned. For example, if you enter a part number, all default representations on the part

appear. If you enter a CAD document number, all default representations for the CAD document appear.

PART DETAILS



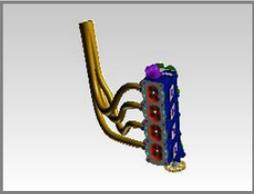
<b>Number</b>	WCDS000079
<b>Name</b>	01-5121541.d prt
<b>Version</b>	A.3 (Design)
<b>State</b>	In Work

DESIGN FILES

 iges_01-5121541.d prt.i	 pdf_01-5121541.d prt.p	 step_01-5121541.d prt.s
---	--	---

- **View Parts List:** On the tailoring page for this app, select the **Include Drawings for Line Items** checkbox to enable the **File Formats** field. The **Drawing** column on the **Parts List** table provides a link from which you can launch the drawing. The file format is based on the order you specify in the **File Formats** field.

**PART DETAILS**



**Number** WCDS000159  
**Name** 01-2\_head.asm  
**Version** A.1 (Design)  
**State** In Work



**PARTS LIST**

Number	Name	Version	State	Quantity	Unit	Drawing
WCDS000656	01-2_cam_bracket.prt	A.2 (Design)	In Work	8.00	ea	<a href="#">View</a>
WCDS000292	cyl_head.prt	A.1 (Design)	In Work	1.00	ea	
WCDS000083	wtr_outlet_rt.prt	A.1 (Design)	In Work	1.00	ea	
WCDS000646	01-2_chain.prt	A.1 (Design)	In Work	1.00	ea	
WCDS000612	01-2_valve_intake.prt	A.1 (Design)	In Work	8.00	ea	
WCDS000666	01-2_cam_exhaust.prt	A.1 (Design)	In Work	1.00	ea	
WCDS000314	01-2_retainer_spring.prt	A.1 (Design)	In Work	16.00	ea	
WCDS000313	valve_exhaust.prt	A.1 (Design)	In Work	8.00	ea	
WCDS000033	01-2_valve_shim.prt	A.1 (Design)	In Work	16.00	ea	
WCDS000310	01-2_valve_lifter.prt	A.2 (Design)	In Work	16.00	ea	<a href="#">View</a>
WCDS000244	01-2_keeper.prt	A.2 (Design)	In Work	32.00	ea	<a href="#">View</a>

## App-specific Tailoring: View Part Properties, Parts List, and Part Structure

The tailoring pages for **View Part Properties**, **View Parts List**, and **View Part Structure** include additional fields for data shape and mashup. These field are explained in [Advanced App Tailoring on page 56](#).

## View Document Tailoring

The following table describes the fields available in the **View Document Tailoring** window.

Field	Description
<b>Search Results Display Limit</b>	Maximum number of objects that display in the <b>Search Results</b> for the app.
<b>Recent Searches Display Limit</b>	Maximum number of objects that display in the <b>Recent Searches</b> for the app.
<b>Additional Attributes Data Shape</b>	Data shape used in ThingWorx Composer that controls the display of the document attributes in the Additional Attributes section of the app. For more information, see <a href="#">Advanced App Tailoring on page 56</a> .

Field	Description
<b>Recent Searches Data Shape</b>	Data shape used in ThingWorx Composer that controls the display of the recent searches in the app. For more information, see <a href="#">Advanced App Tailoring on page 56</a> .
<b>Download on Search</b>	Determines whether or not the document is automatically downloaded and available to the user to open.   <b>Note</b> You may need to modify the pop-up blockers settings in your browser accordingly.
<b>Download Format</b>	Determines whether the representation (such as a PDF) or the native file (such as a Word document) is opened.
<b>Mashup</b>	A link to ThingWorx Composer from which you can view the app mashup.
<b>Allow Partial Matches in Search</b>	Enable search results to return more than exact matches.
<b>Search on Attribute</b>	Select whether to search on the object number, name, or both.
<b>Enable Search Filter</b>	Select the checkbox to enable the <b>Select Filter Type</b> field from which you can select a filter to be available when users search from the apps.
<b>Select Filter Type</b>	This field is available when the <b>Enable Search Filter</b> checkbox is selected.  Select a filter to be available when users search from the apps: <ul style="list-style-type: none"> <li>• <b>Saved Search:</b> The saved searches display in a drop-down list in the app search. Select saved searches available in the Windchill site context. For additional information, see the following section “Best Practices for Saved Searches.”</li> <li>• <b>Latest:</b> The lifecycle state and view by which users can filter their search. You can select multiple options from the list. Note that the search results return the latest iterations of all object revisions. To return only the latest iteration of the latest revision, create a save search as described in the following section “Best Practices for Saved Searches.”</li> <li>• <b>Version:</b> A <b>Version</b> field displays in the app search, enabling users to filter their search by an object version, for example A.1. Users must also enter information in the search field.</li> </ul>

---

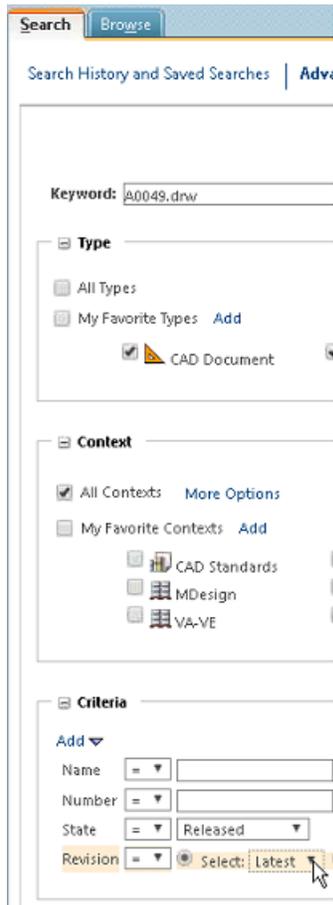
## Best Practices for Saved Searches

Consider the following information when using saved searches in PTC Navigate.

- To avoid access control issues, use saved searches from the Windchill site context. Other contexts, such as products or libraries, may restrict user access to the saved search.
- If your Windchill server is configured to index document content using Solr, saved searches return results matching indexed document content.
- PTC Navigate apps and their tailoring options filter the results of a saved search. Consider the following examples:
  - Although a saved search may include multiple object types, the **View Document** app returns documents only.
  - Although a saved search may include multiple object types, the **Use CAD Documents as Parts** tailoring option filters the app results for CAD documents only.
  - The **Search Results Display Limit** tailoring option limits the results of a saved search.

As a best practice, configure saved searches specific for the object type of each app because a search for multiple object types clutters the **Search Results Display Limit** and could result in fewer search results of the desired object type. For example, if you want the app to search only CAD documents but the saved search also includes parts, you may not see all of the CAD documents in your app search if the **Search Results Display Limit** was set less than the results for a search of both parts and CAD documents.

- To search for only the latest revision of an object, use one of the following methods:
  - Create a saved search in Windchill that specifies the **Revision as Latest**. You can also set a **State** for your saved search, for example, return only the latest released revisions as shown in the example below:



- Set the Windchill preference **Latest Version Search** to **Yes**.

**Preference Management** Standard

Name ↑	Value	Description
Project Planning		
Promotion Process		
PSE		Product Structure Explorer
Reports		
Revise		Revise Preferences
Save As		Save As operation preferences
Search		
All Applicable Object Types Search	Abstract Specification,...	List of types shown in Type Picker in Search
Enable Index Search	Yes	Choose to perform an index search when t
Enable Search Facets	No	Enables facets on the Search results page.
Excluded Part Subtypes		Configures the part subtypes that are exclu
Global Search Default Types List	Part, Document, CAD ...	This preference specifies the object types d
Implicit Wildcards for Search in Structures	Yes	Determines whether wildcards are implicit i
Latest Version Search	Yes	Search for the latest version of objects.
Recently visited Context Picker list size	5	Recently visited Context Picker list size

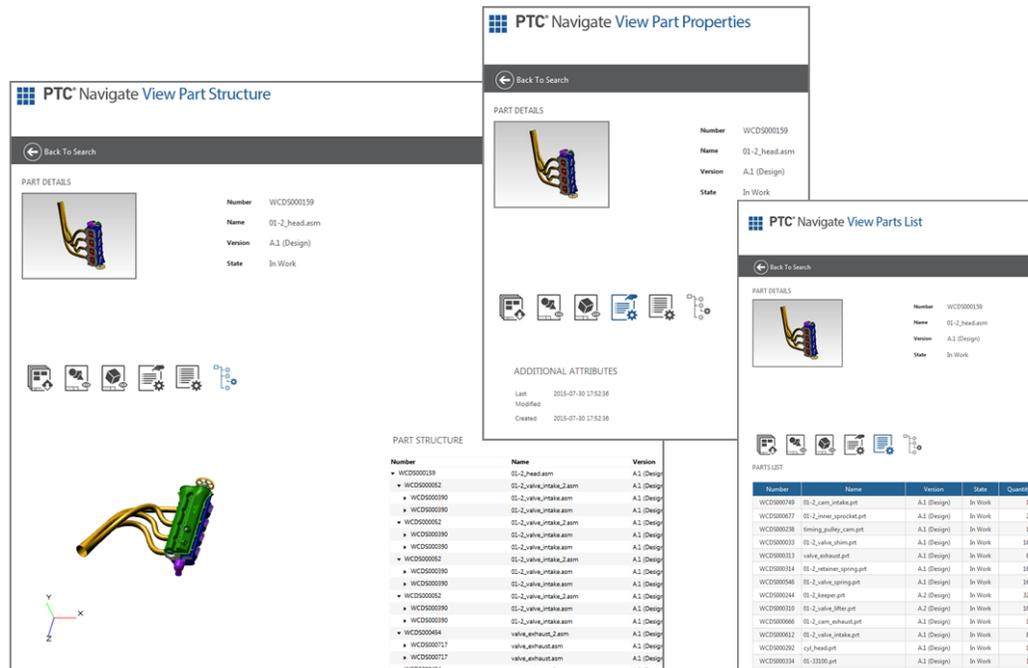
# Advanced App Tailoring

You can choose to use ThingWorx Composer to configure the apps beyond the options described in [Basic App Tailoring on page 47](#). Experience with ThingWorx Composer is recommended. The sections to follow demonstrate some common advanced tailoring options.

Additional customization information is available in the following appendix [Developing a Customized Application on page 91](#).

## Data Shape and Mashup Information

The **View Part Structure**, **View Part Properties**, and **View Parts List** apps also contain an additional ThingWorx mashup containing the data shape for an additional component. For example, the **View Part Properties** app has **Additional Attributes**.



The name of the app-specific data shape can be found on the tailoring page specific to the app. For example, the **View Parts Properties** tailoring page tells you that the **Additional Attributes Data Shape** can be found in ThingWorx Composer as the following entity: **Example1PartPropertiesAccessAppDataShape**. To launch ThingWorx Composer, click the link available from the **Mashup** field.

Additional Attributes Data Shape:

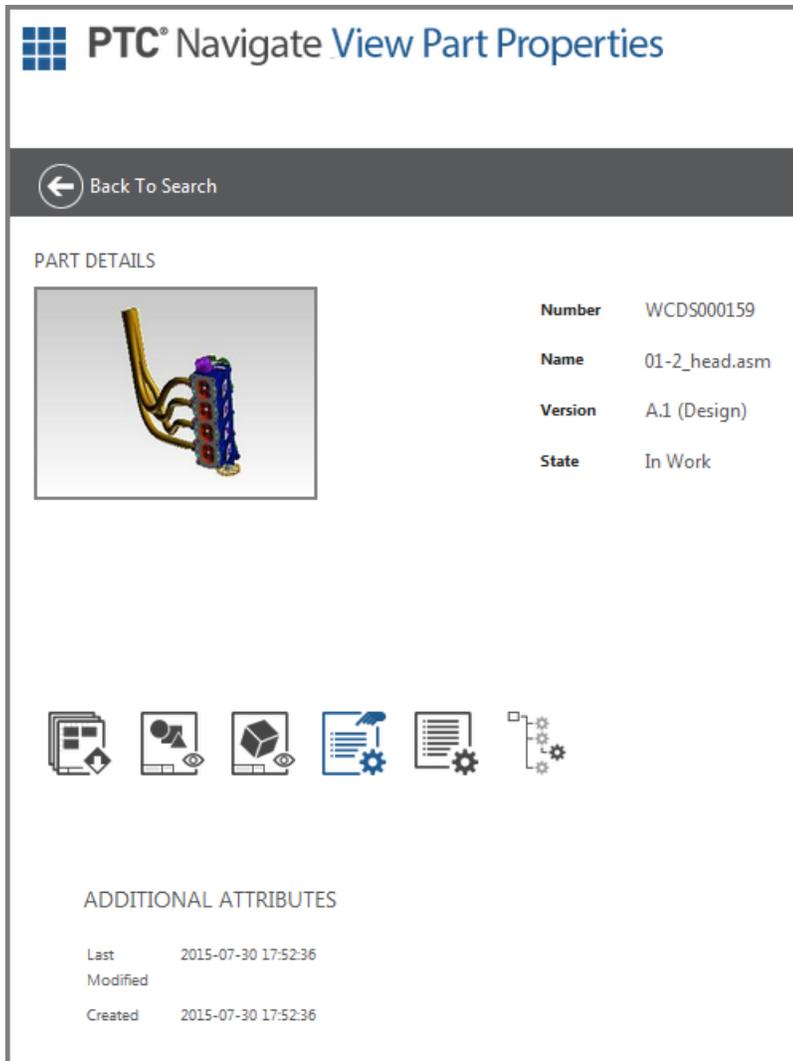
Mashup:

To tailor the additional attributes shown in the mashup you must edit this data shape and configure the **PartProperties** widget in the mashup.

## Modifying Properties in App Data Shapes

Your site can tailor the information shown in the app-specific mashup as described previous section. Sites cannot tailor the standard attributes (**Number**, **Name**, **Version**, and **State**) that display at the top of app details pages.

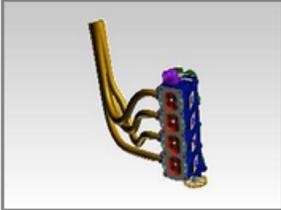
The following example demonstrates how to add, delete, or re-order properties in the **Additional Attributes** of the **View Part Properties** app.



PTC® Navigate **View Part Properties**

← Back To Search

PART DETAILS

	<b>Number</b> WCD5000159
	<b>Name</b> 01-2_head.asm
	<b>Version</b> A.1 (Design)
	<b>State</b> In Work

Icons: [Grid], [Microphone], [Cube], [List with Gear], [List with Gear], [Tree with Gear]

ADDITIONAL ATTRIBUTES

Last Modified	2015-07-30 17:52:36
Created	2015-07-30 17:52:36

1. Select the edit icon  for **View Part Properties**.
2. From the **View Part Properties Tailoring** page, make note of the following information: mashup, data shape and widget.

Additional Attributes Data Shape:  To tailor the additional attributes shown in the mashup you must edit this data shape and configure the **PartProperties** widget in the mashup.

Mashup:

3. Click the **Mashup** link and complete the steps in the “Editing the Mashup” and “Editing the Data Shape” sections below.

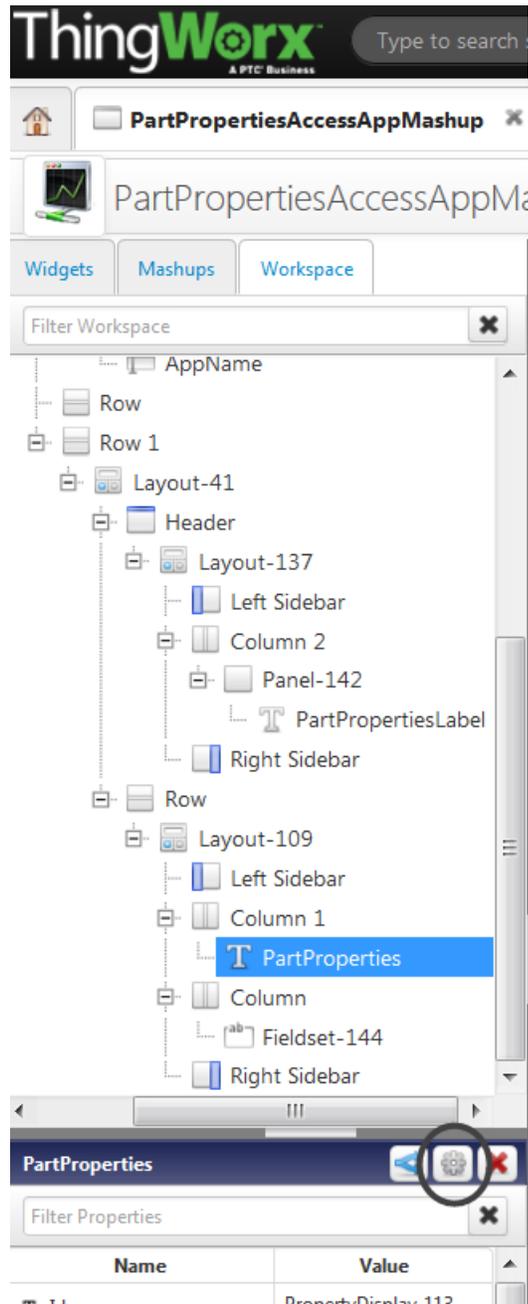
## Editing the Mashup

1. From the home page, search for and select the mashup listed on the tailoring page, for example, PartPropertiesAccessAppMashup.

The screenshot shows the ThingWorx interface. At the top, there is a search bar with the text 'Type to search system' and a '+ New Entity' button. Below the search bar, there is a navigation menu with categories like 'MODELING' and 'VISUALIZATION'. The 'Mashups' category is selected. In the main area, there is a search bar with the text 'PartPropertiesA' and a 'Filtering by: Exclude System Objects' button. Below the search bar, there is a table with the following data:

View	Name	Description
	PartPropertiesAccessAppMashup	A sample mashup that demonstrates

2. On the mashup page, click the **Workspace** tab.
  - a. Select the widget you made note of on the PTC Navigate tailoring page, for example, PartProperties.
  - b. Click the Configure Widget icon in the pane below.



3. The **Configure Widget** window lists the properties available for display. For this example, select **Quantity**. From this window, you can also make the following modifications:
  - Drag-and-drop the properties to change the order in which they display in the app.
  - Clear a checkbox to remove the property from the app.

---

### Note

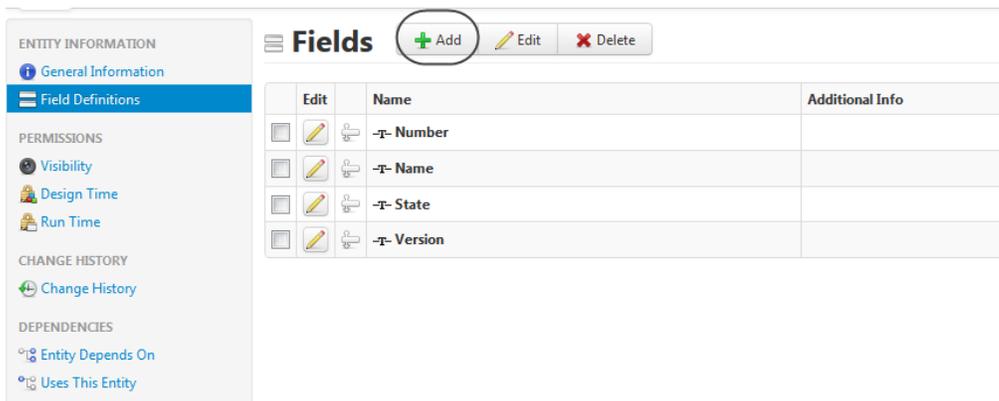
Attributes created by your site are not available in this window by default. For information on including site-specific attributes in your app, see [Configuring Windchill Attributes as App Properties on page 65](#).

---

4. Click **Done**.
5. Save the mashup.
6. Complete the steps in the next section “Editing the Data Shape.”

### Editing the Data Shape

1. From the ThingWorx Composer home page, search for and select the data shape identified on the tailoring page, for example, Example1PartPropertiesDataShape.
2. Under **Entity Information**, select **Field Definitions** and then click **Add**.



Edit	Name	Additional Info
	-T- Number	
	-T- Name	
	-T- State	
	-T- Version	

3. A **New Field** pane appears.
  - a. In **Name**, enter the internal name for the attribute found in the Windchill Type and Attribute Management utility.

---

### Tip

For part apps, you could use attributes for part, part master, or part usage types.

---

- b. In **Base Type**, select the option associated with the attribute you entered in the **Name** field. For example, if you entered `Quantity`, you would select **Number** for the base type.
4. Click **Done**.

---

 **Tip**

To use a property label other than the internal name entered in this procedure, see [Modifying a Property Label on page 61](#).

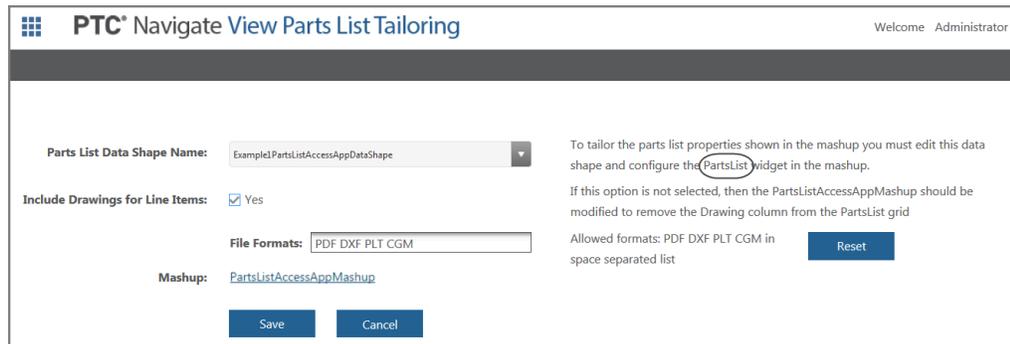
---

## Modifying a Property Label

To modify the text of a property label, use ThingWorx Composer to create or add localization tokens. For more information, see [Localizing Labels and Fields in Your Application on page 92](#).

This example uses the **View Parts List** app.

1. Click the edit icon  for the **View Parts List** app.
2. From the **View Parts List Tailoring** page, make note of the following.
  - Mashup: PartsListAccessAppMashup
  - Widget: PartList



PTC\* Navigate View Parts List Tailoring Welcome Administrator

Parts List Data Shape Name:

Include Drawings for Line Items:  Yes

File Formats:

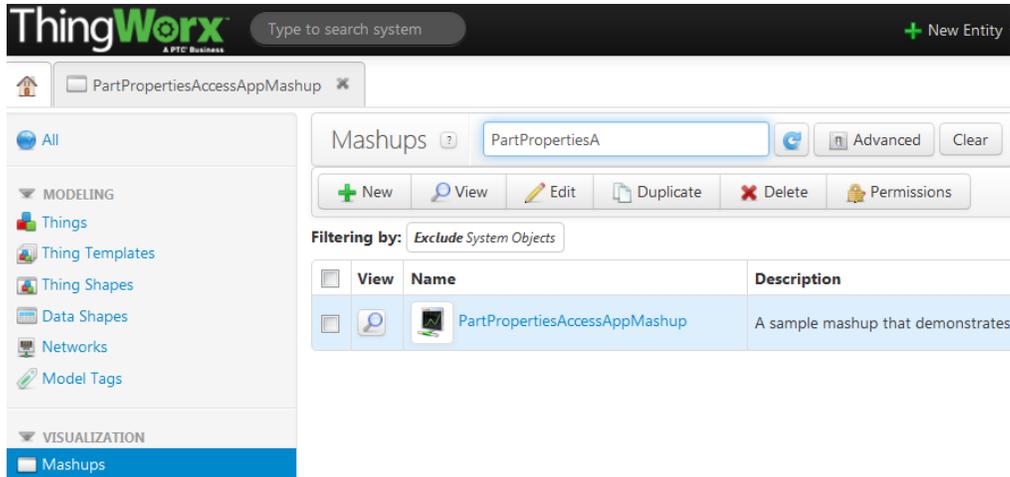
Mashup: [PartsListAccessAppMashup](#)

Allowed formats: PDF DXF PLT CGM in space separated list

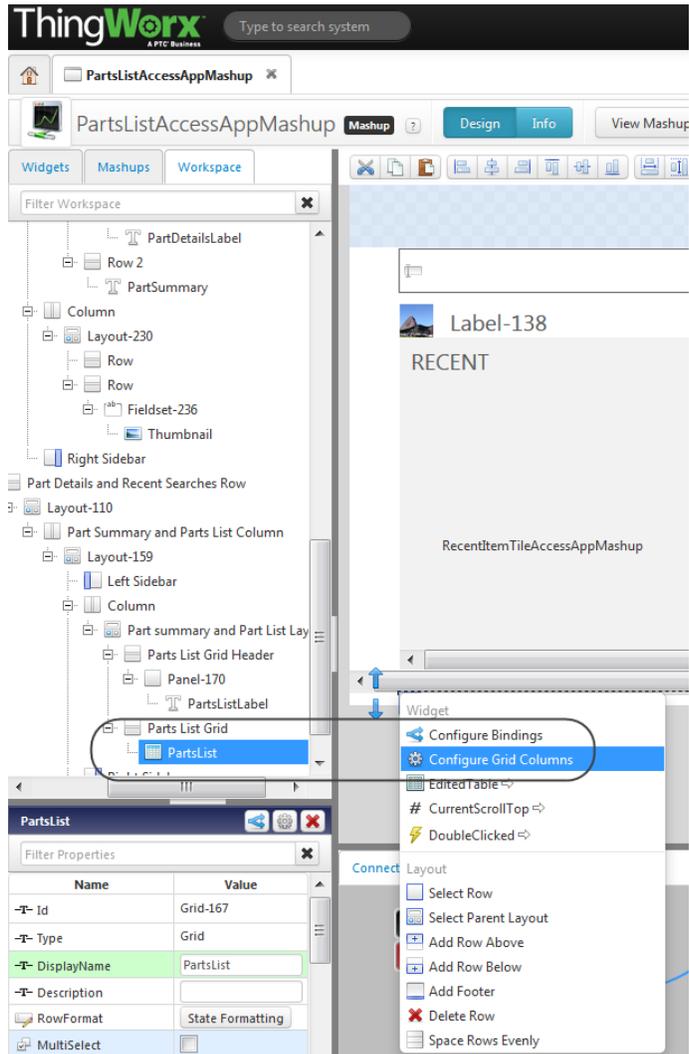
To tailor the parts list properties shown in the mashup you must edit this data shape and configure the **PartsList** widget in the mashup. If this option is not selected, then the PartsListAccessAppMashup should be modified to remove the Drawing column from the PartsList grid

3. Click the **Mashup** name to launch ThingWorx Composer.

4. From the home page, search for mashup listed on the tailoring page, for example, PartsListAccessAppMashup.

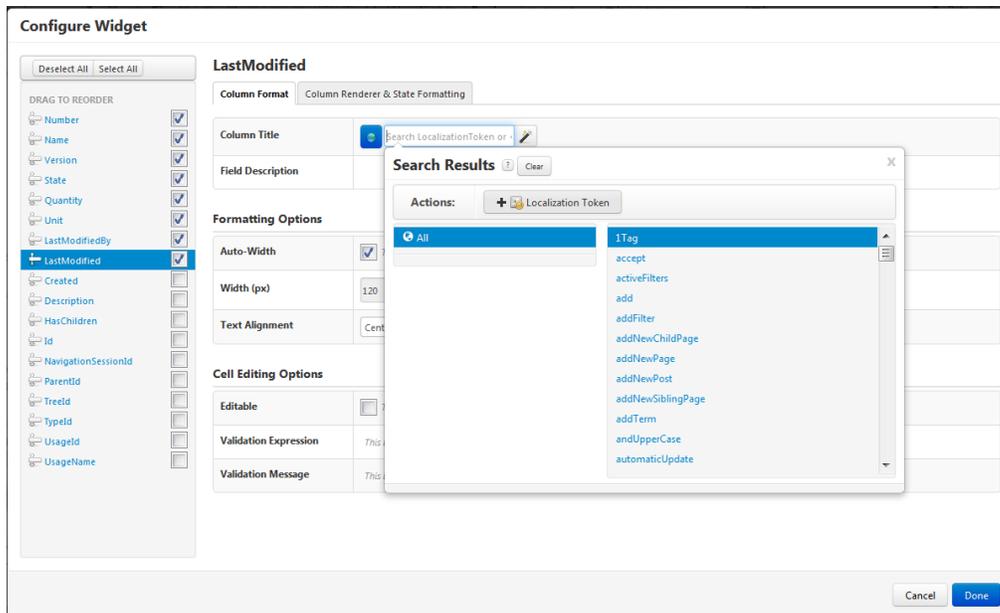


5. On the mashup page, click the **Workspace** tab.
  - a. Navigate to the widget listed on the tailoring page, for example, PartList.
  - b. From the action drop-down for the grid widget, select **Configure Grid Columns**.



6. In the **Configure Widget** window, select the checkbox for the property label you want to modify. For this example, select LastModified.

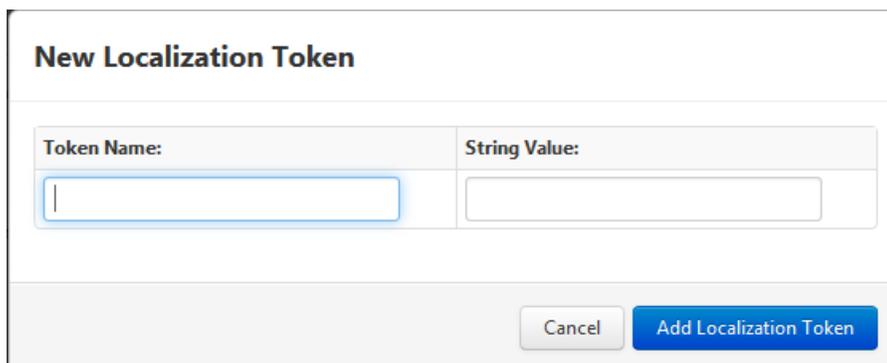
- Click the wand icon  next to the **Column Title**.



 **Tip**

If the wand icon does not appear, it means the localization has already been set for the property. To change it, click the “x” to delete it and the wand icon appears.

- Click **Localization Token**.
- The **New Localization Token** window opens.
  - Token Name:** Internal placeholder for strings replaced at runtime for localization.
  - String Value:** Label that displays in the app.



10. Click **Add Localization Token**.

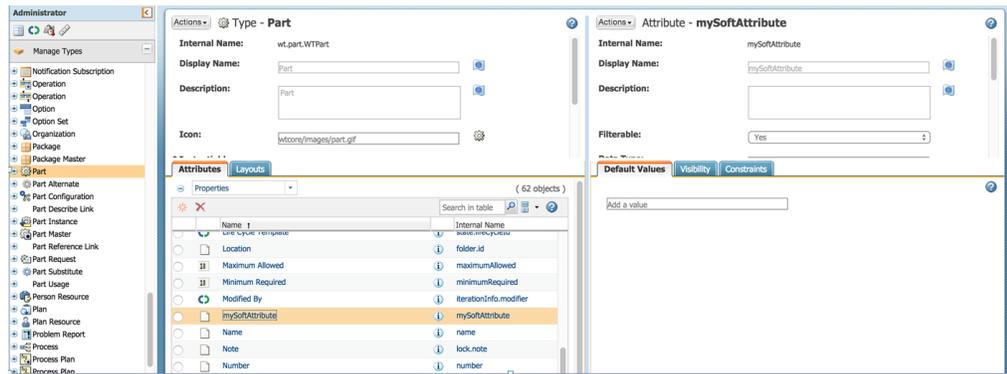
11. Save the mashup.

## Configuring Windchill Attributes as App Properties

Your site can tailor the information shown in the app-specific mashup as described previous section. Sites cannot tailor the standard attributes (**Number**, **Name**, **Version**, and **State**) that display at the top of app details pages.

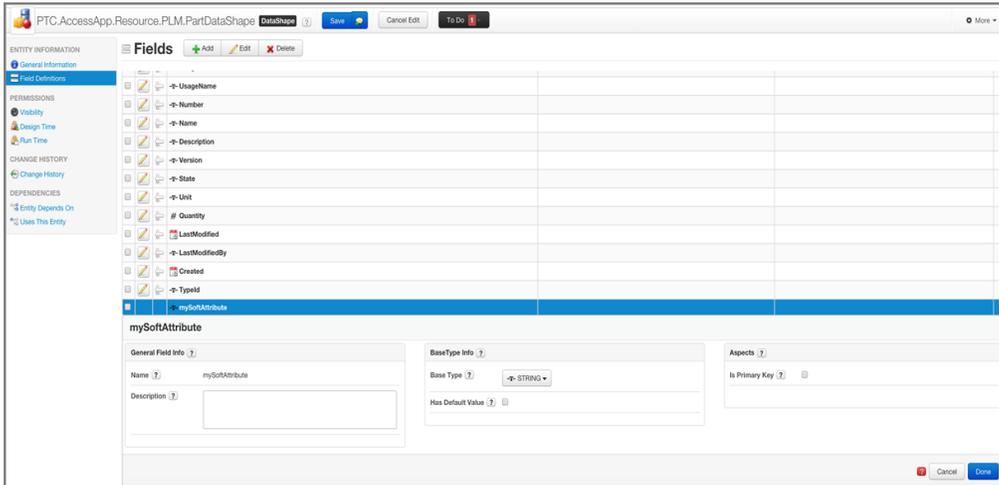
This example uses the **View Parts List** app.

1. In the Windchill Type and Attribute Management utility, identify **Internal Name** for the attribute you want to display on the app. For this example, select the `mySoftAttribute` attribute. If you wanted to use the Modified By attribute, you would need “iterationInfo.modifier.”

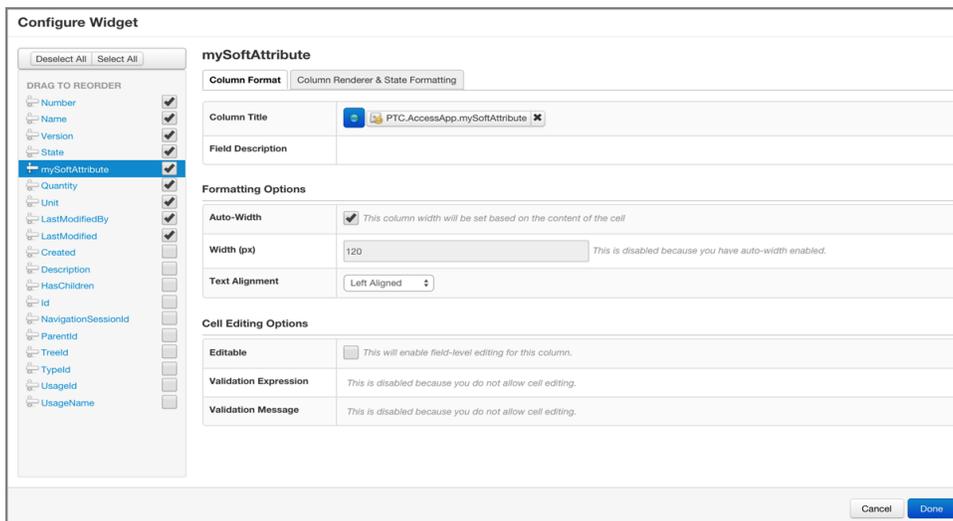


2. In ThingWorx Composer, add the `mySoftAttribute` attribute as field definitions of the following data shapes. For more information, see Editing the Data Shape section under [Modifying Properties in App Data Shapes on page 57](#).

- `PTC.AccessApp.Resource.PLM.PartDataShape`
- `Example1PartsListAccessAppDataShape`



3. In ThingWorx Composer, open the **PartsListAccessAppMashup**.
4. Select the **PartsList** grid widget, which will modify the display for the **Parts List** table in the **View Parts List** app:
  - a. Select **Configure Grid Columns**.
  - b. Select the `mySoftAttribute` attribute from the list of available attributes and complete the following steps as required:
    - i. Drag-and-drop columns to change the order of the attributes.
    - ii. Create or add a localization token.
    - iii. Specific a format for the data.



- c. Ensure the remainder of the selected attributes match the field definitions in **Example1PartsListAccessAppDataShape** and are configured as described in the previous step.
5. Save all changes.

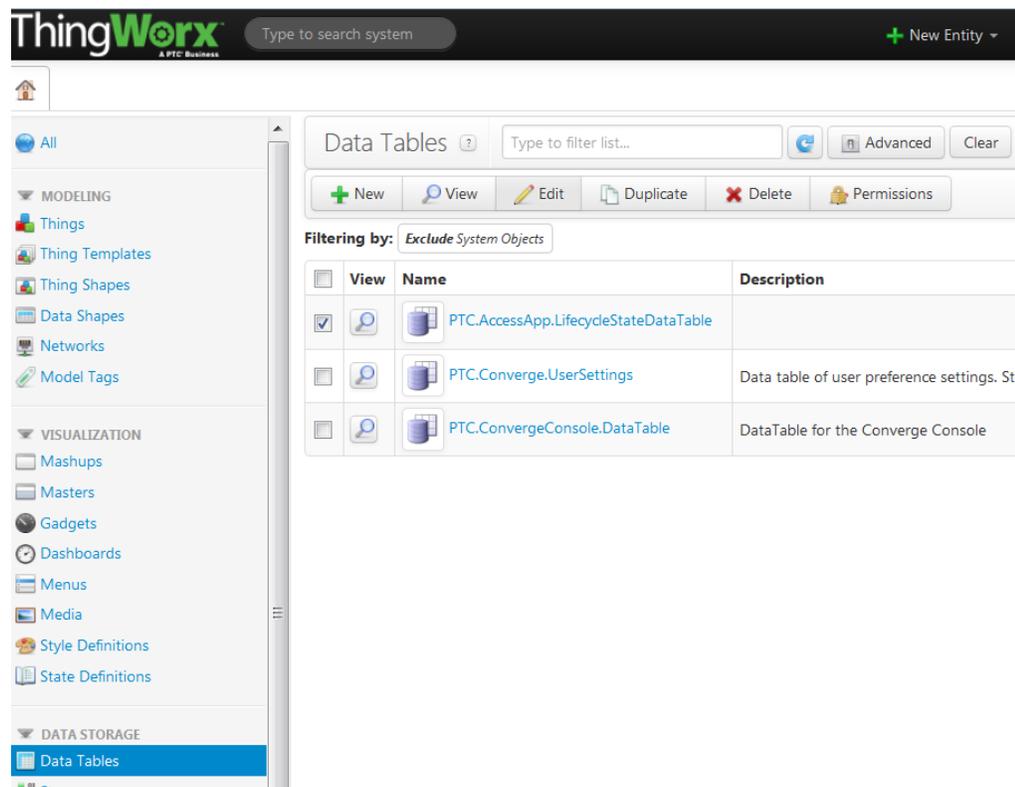
## Adding a New Windchill Lifecycle State

The following Windchill lifecycle states are available in PTC Navigate by default:

- Released
- Cancelled
- In Work

Use the following procedure to add additional lifecycle states in PTC Navigate.

1. In ThingWorx Composer, under **Data Storage**, select **Data Tables**.
2. Select **PTC.AccessApp.LifecycleStateDataTable** and click the **Edit** button.



3. Under **Entity Information**, click **Services**.
  - To add one new lifecycle state, locate the **AddDataTableEntry** service name and click the **Test** button.
  - To add multiple new lifecycle states, locate **AddDataTableEntries** service name and click the **Test** button.
4. Under **values**, click the **Edit** button.

The **Editing Infotable** window opens.
5. Click the **Add** button.
  - a. Under **name**, enter a valid Windchill lifecycle state value.

- b. Under **description**, enter the display name for the lifecycle state.
6. Click the **Save** button.
7. Click the **Execute Service** button.

An id appears when the service execution completes successfully. Close the **Test Service** window.

8. To review newly added lifecycle states, execute the service for **GetDataTableEntries**. Then go to the app tailoring page to see the new options in the **Lifecycle State** field.

**PTC<sup>®</sup> Navigate Common Tailoring for Part Apps**

Recent Searches Data Shape: WindchillDemoRecentSearchesPerUser  To tailor the fields shown in the recent searches you must edit this data shape.

Use CAD Documents as Parts:  Yes

Allow Partial Matches in Search:  Yes

Search on Attribute:  Number  Name  Number or Name

Enable Search Filter:  Yes

Select Filter Type:  Saved Filter  Saved Search  Latest  Version

**Lifecycle State:**  Canceled  
In Work  
My State  
Released

**View Name:**  EBOM  
MBOM  
Service

## Configuring Non-Windchill Attributes as App Properties

This section describes how to augment the data shown in the PTC Navigate apps with data that comes from another enterprise system. For example, fields from an ERP or MES system can be added to PTC Navigate apps to view the Windchill data alongside data from these other systems.

This section describes how to create a *contributing resource provider*. This resource provider is intended to be used in conjunction with the ontology-defined resource providers. An *ontology-defined resource provider* provides access to resources where the available fields come from the ontology data shape. A contributing resource provider augments the resources provided by the ontology-defined resource provider with additional fields that come from some other source.

---

PTC Navigate comes with ontology-defined resource providers that provide access to Windchill resources. This section contains instructions on how to create a contributing resource provider that augments these ontology-defined resources with fields from non-Windchill systems.

The **SAPDemoPartResourceProvider** thing is an example contributing resource provider that is included with PTC Navigate. Its configuration rules are disabled so that it is not used unless its configuration records are enabled.

Follow these steps to create your own contributing resource provider to enable additional fields that can be included in the PTC Navigate apps.

1. Create a resource provider.

Follow the ThingWorx Converge documentation to create a resource provider thing. In its most simple form, this can be done in the ThingWorx Composer following these steps:

- a. Create a new thing using the **PTC.Resource.ResourceProviderTemplate** thing template.
- b. Add the necessary attribute mappings to the configuration of the thing.
- c. Override the FindResources service. See below for details on how to implement this service.
- d. See the ThingWorx Converge documentation for more details on how to implement a resource provider.

2. Implement the contributor thing shape.

Add the **PTC.AccessApp.Resource.ContributingProviderThingShape** thing shape to the list of thing shapes that the resource provider implements. This is what makes an ordinary resource provider a contributing resource provider. This thing shape adds the **ContributingResourceProviderConfig** property to the resource provider.

3. Configure the resource provider thing.

Add rows to the **ContributingResourceProviderConfig** property on the resource provider thing. The configuration rules control under which conditions this resource provider will be invoked to provide additional fields to the resource. Each row consists of the following:

- **ResourceProviderName**

The name of the resource provider thing to which this contributing resource provider contributes fields. Leave this blank to contribute fields to all resource provider things.

- **ServiceName**

---

The name of the service on the resource provider thing to which this contributor will contribute fields. For example, to contribute fields to the `Get` service, set this field to `Get`. Leave this field blank to contribute fields to all services on the resource provider specified by the **ResourceProviderName**.

- **JoinFieldName**

The name of the field from the resource that is provided in the query argument of the `FindResources` service. The query is a JSON object as follows:

```
{
  filters: {
    type: 'EQ',
    fieldName: <join-field-name>,
    value: <field-value>
  }
}
```

- **AddContributorInfo**

This boolean flag indicates whether information about this contributing resource provider is added to the **ContributorInfo** field on the resource that is returned to the client. Note that for a client to see this field, it must exist in the resource's data shape and must be in the data shape the client requests when it calls the ThingWorx Converge resource manager.

- **Enabled**

This Boolean flag indicates whether this rule is enabled or not. If a configuration entry is disabled then the contributing resource provider is not invoked and no fields are contributed to the resource.

4. Implement the `FindResources` service.

Override the `FindResources` service on the contributing resource provider. This service is passed a query which specifies which resources to fetch. A typical implementation extracts the necessary data from the query and invokes an endpoint in the external system to get the necessary data. It then adds a row to the `Infotable` that contains the data from the external system.

5. Add contributed fields to the ontology-defined resource data shape.

Edit the ontology-defined resource data shape. This is the data shape that defines the ontology resource. Add fields to it.

6. Add contributed fields to the client's data shape.

Edit the data shape that is bound to the UI widget. The tailoring mashups give the name of the client data shapes. Edit the data shape and add the fields that

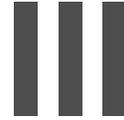
---

are provided by the contributing resource provider and that you want to make available to the UI.

7. Update the mashup.

Edit the configuration of the widgets in the mashup to make the additional fields visible.





# PTC Navigate View ALM App Extension

The View Requirements app, a PTC ALM app enables you to view various types of PTC Integrity documents. A case insensitive text search and the ability to configure metadata is also provided with this app.

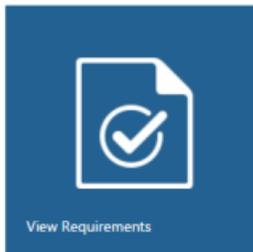
---

# 10

## PTC Navigate View ALM App Extension Product Overview

PTC Navigate View ALM App Extension provides an app that allows users to quickly and easily access specific information stored in PTC Integrity Lifecycle Manager. The **View Requirements** app is available with this release.

### PTC® Navigate



This app provides a search field in which you enter document ID of the document you want to view. The document type that can be viewed in the app can be modified through app tailoring. For more information, see [Tailoring PTC Navigate View ALM App Extension on page 83](#).

This landing page is a ThingWorx mashup available at the following URL:

```
http://<host>:<port>/Thingworx/Runtime/  
index.html#master=PTC.AccessApp.Master&mashup=  
LandingPageAccessAppMashup
```

---

** Best Practice**

PTC Navigate apps were designed for a screen resolution of 1280x1024.  
Results on other resolutions may vary.

---

---

# 11

## Installing PTC Navigate View ALM App Extension

The following topic provides an overview of the process to download and install the required components for PTC Navigate View ALM App Extension.

### PTC Navigate View ALM App Extension Product Support

The section details the product support for PTC Navigate View ALM App Extension:

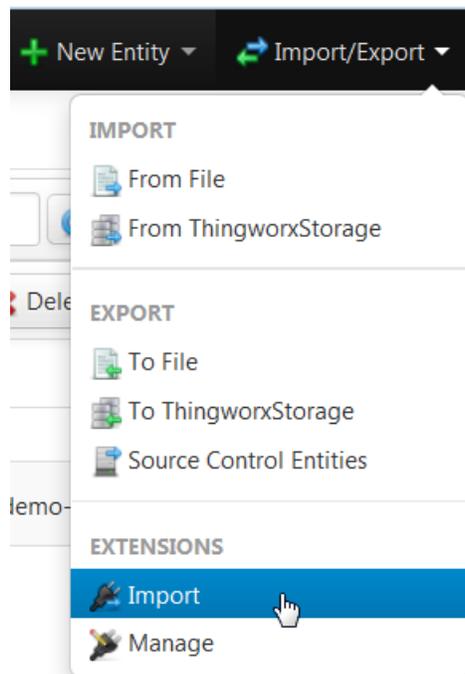
- Integrity at one of the following release levels:
  - 10.8 or later
- Apache Tomcat 8.0.30 or later
- Browser support
  - Firefox 35 and later
  - Chrome 44
  - Safari 6.1.6 and later

---

## Importing PTC Navigate View ALM App Extension Files

To install the extensions included with the PTC Navigate View ALM App Extension, complete the following steps:

1. Download the **PTCNavigateViewALMAppextension.zip** file from the PTC Smart Connected Applications section of the PTC Software Downloads page and unzip the contents to a folder on your machine.
2. From ThingWorx Composer, navigate to **Import/Export** ► **Import**:



3. In the **Import Extensions** window, navigate to `TWX_Integrity_LM_Connector_ExtensionPackage.zip`.

---

### Note

This release of PTC Navigate View Requirements is compatible only with the version of ILM Connector extension which is packaged. It will not be compatible with the version of ILM Connector extension which is packaged with PTC Integrity Extension for ThingWorx.

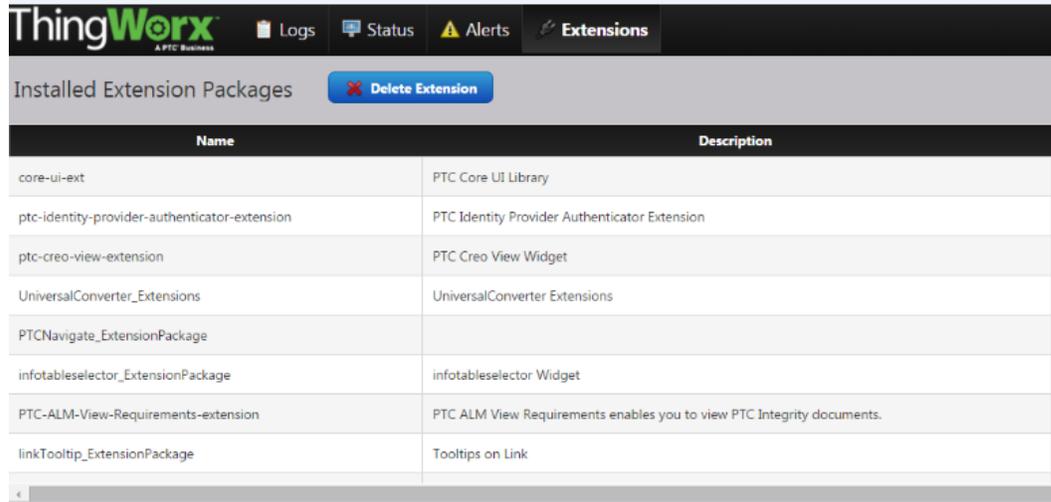
If the PTC Integrity Extension for ThingWorx needs to be imported, it should be done on a separate Thingworx installation different from the one used for PTC/Navigate.

---

4. Click **Import** and refresh the window if prompted.

5. Again select **Import/Export ► Import**.
6. Navigate to `PTC_Navigate_View-ALM_App_Extension.zip` and click **Import**.

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
core-ui-ext	PTC Core UI Library
ptc-identity-provider-authenticator-extension	PTC Identity Provider Authenticator Extension
ptc-creo-view-extension	PTC Creo View Widget
UniversalConverter_Extensions	UniversalConverter Extensions
PTCNavigate_ExtensionPackage	
infotableselector_ExtensionPackage	infotableselector Widget
PTC-ALM-View-Requirements-extension	PTC ALM View Requirements enables you to view PTC Integrity documents.
linkTooltip_ExtensionPackage	Tooltips on Link

# 12

## Configuring PTC Navigate View ALM App Extension

### Configuring ThingWorx for Authentication

You can sign into ThingWorx using either single sign-on authentication solution or ThingWorx-based authentication.

If you are using a single sign-on authentication solution, Windchill is used as the identity provider for ThingWorx. Hence, to access a ThingWorx URL, your identity must exist in ThingWorx, Windchill, and PTC Integrity. For more information, see [Configuring ThingWorx Single Sign-On when Connecting to Windchill](#) on page 38.

In absence of single sign-on, you need to use ThingWorx-based authentication. In this case, your user identity must exist in both ThingWorx and PTC Integrity. If users are in LDAP, you must configure LDAP for authentication. For details, see the topic Directory Services Authentication in the ThingWorx Help Center.

### Configuring for Impersonation

Once you are logged into PTC Navigate, authentication to PTC Integrity is done by impersonation where administrator permissions are required for impersonation. In this case, the Integrity Connector retrieves documents from PTC Integrity by impersonating the logged in ThingWorx user. Hence, the user specified in the Connector properties must have impersonation permissions in order to impersonate the current user.

---

Carry out the following steps using the Integrity Administration Client to provide impersonation rights to all users:

1. In Integrity Administration Client, and select **Permissions ► All**. A complete tree view of all ACLs on the server is displayed.
2. Select **ACL ► Create ACL**. The **Select ACL Entries to Add** window opens.
3. In the **ACL Name** field, type `mks:impersonate:group:everyone`.
4. From the **Principal** list, select **Administrator**.
5. In the **Permissions** list, click the **Impersonate** checkbox.
6. Click **OK** to accept the changes.

## Configuring the PTC Integrity Server Properties

Carry out the following steps in ThingWorx Composer to configure the PTC Integrity Server properties:

1. In ThingWorx Composer, click the **IntegrityServer** thing.
2. Under **ENTITY INFORMATION**, select **Properties**.
3. Enter appropriate information in the following fields:
  - port
  - hostname
  - username
  - password
4. Restart the Tomcat web server.

## Configuring PTC Integrity Lifecycle Manager

This configuration involves running the **UpdateConnectorConfiguration** service, which ensures proper display of rich text fields in the View Requirements app. You must run this service only once and the ThingWorx user executing this service must have administrative rights in PTC Integrity.

Complete the following steps to configure the ILM Connector:

1. In the ThingWorx Composer, click the **IntegrityServer** thing.
2. Under **ENTITY INFORMATION**, select **Services**.
3. Click **Test** to run **UpdateConnectorConfiguration** service.
4. Close the **IntegrityServer** thing.

---

## Configuring the TimerThing for Deleting Media Entities

Media entities are images or OLE objects contained within a document. When you load a document for viewing in the View Requirements app, all images and OLE objects in the document are stored in ThingWorx.

The administrator can configure the TimerThing such that all media entities created while loading a document are deleted by default after a specified time interval. This time interval in milliseconds is set for the **Update Rate** field as follows:

1. In the ThingWorx Composer, under **Things**, click the **PTCRODocViewMediaEntityCleanUpTimerThing** thing. The mashup for the thing opens.
2. Under **Configuration**, enter a value in milliseconds in the **Update Rate** field.  
Based on the interval in **Update Rate**, media entities that are created by loading a document are deleted.
3. Click **Save**.

## Modify ThingWorx Run Time Permissions

### Configuring for Role-based Permissions

By default, permissions for View Requirements app non-administrative users are set to read only. To manage user access, modify the permissions in **Run Time** for entities in ThingWorx.

The following procedure describes the permissions to set in order to make the View Requirements app visible to a given group.

1. Create user group (treated as user role) and add specific users to this group.
2. Under **Run Time** permission, give **Property Read** and **Service Execute** access to each thing that corresponds to the View Requirements app for which you want the user to have access.
3. Set all needed permissions for entities as described in the table below. The descendent things and thing templates related to View Requirements app inherit the permission setting.

Entity	Permission	Permission Type
PTCRODocViewAccessAppNavigateThing	Run Time	Property Read Service Execute
AppCollectionsConsolidator	Run Time	Property Read Service Execute
PTCRODocViewAppCollectionProviderThing	Run Time	Property Read Service Execute

### Configuring the Access to Administrator Tailoring

Tailoring options enable administrators to configure the type of content that is displayed while viewing a document in the View Requirements page. In order to perform this configuration, the edit icon for the app on the PTC Navigate product page must be visible to administrators to gain access to tailoring options.

Administrators can also grant permissions to user groups for accessing tailoring options. Once permissions are granted, all users in the group can see the edit icon for the View Requirements app on the PTC Navigate product page.

To grant permissions to user groups, the administrator must configure the **PTCRODocViewAccessAppThing** in ThingWorx as follows:

1. In the ThingWorx Composer, under **Things**, click the **PTCRODocViewAccessAppThing** thing. The mashup for the thing opens.
2. Under **Permissions**, select **Run Time** type of permission.
3. Grant Allow permissions to **Property Read** and **Service Execute**.
4. In **Group or User**, add the required user group.
5. Click **Save**.

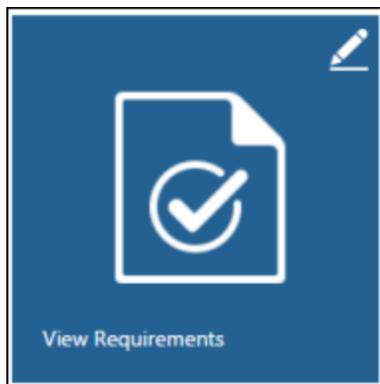
# 13

## Tailoring PTC Navigate View ALM App Extension

### Basic App Tailoring

PTC Navigate provides administrators basic tailoring capabilities related to the app display and function. When you access the landing page as an administrator, edit icons appear for the apps. This page is available at the following URL:

```
http://<host>:<port>/Thingworx/Runtime/  
index.html#master=PTC.AccessApp.Master&mashup=  
LandingPageAccessAppMashup
```



An administrator can use options on the following tailoring page to customize the display of document fields in the View Requirements app interface:

**Specify the fields to be displayed in the View Requirements mashup**

**Document Type \***

**Document Field \***

**Sections Field**

\*Required

[Save Configuration](#)

**CONFIGURATIONS**

Document Type	Item Type	Document Field	Sections Field
Input Document	Input	Text	Text
Requirement Document	Requirement	Text	Text

[Remove](#)

To customize the display in the View Requirements app page, complete the following steps:

1. In the **Document Type** list, select the type of document you want to view from. For example, you can select a Design Document type. Selecting the document type is mandatory.
2. In the **Document Field** list, select the field that represents the main content of your document. For example, you can select a field that stores the design details in a design document.
3. Optionally, in the **Sections Field** list select the field that you can view in the **Sections** view of the document.
4. Click **Save Configuration** to accept the changes. The saved configurations are added to the **CONFIGURATIONS** table.

Note: You can save only a single configurations for a document type .

The columns in this table indicate the following:

- Document Type: Indicates the type of document you have selected.
- Item Type: Indicates the item type node.

- 
- Document Field: Indicates the type of content that is displayed in the **Document** view of the document.
  - Sections Field: Indicates the type of content that is displayed in the **Sections** view of the document.

### **Removing a Configuration**

Select the configuration you want to delete, and click Remove. The configuration is deleted from the **CONFIGURATIONS** table.

---

# 14

## Known Limitations PTC Navigate View ALM App Extension

The following notes describe known limitations for the View Requirements app.

### Entity Not Found Error

Embedded OLE objects are created on loading a document. These objects are deleted through a timer-based service after every specified time interval that is defined by an administrator. Sometimes, when viewing documents in the View Requirements app, you may get the error **Entity Not Found : [<name of the entity>]** on clicking an embedded Word document, PDF file, or an OLE object.

This occurs only when you are viewing a document when OLE objects are being deleted in the background by a timer-based service.

You can resolve this issue by reloading the page so that all the OLE objects get created again.

### Requirement to Update Configuration Table

Configurations in the tailoring page do not get updated automatically, if you modify the Integrity host details on **IntegrityServer** thing. This may lead to errors as these configurations do not correspond to the updated server document fields.

To resolve this issue update the **CONFIGURATIONS** table in the tailoring page so that updated configurations map correctly to the fields that exist on the new Integrity Server.

### Non-English Locales Not Supported

Currently, you can configure only English locale Integrity Servers with the View Requirements app.

---

## **Performance Issues with Large Documents**

When you load a document with more than 10,000 nodes, the performance of the View Requirements app is slower than expected.

## **Large Documents Loading Slowly with Internet Explorer 11.0**

When working with the View Requirements app in Internet Explorer 11.0, large documents may take long to load as compared to other browsers.

## **FVA Field Issues with Integrity Server**

When you have a ALM solution installed Integrity Server, where the FVA field has the Text attribute prefixed with ALM (For example, ALM\_Text), the Integrity Server may not work as expected in View Requirements App 1.0 version.

## **Table Background Color Issue**

The background color of tables is not visible while viewing documents in the View Requirements app.



————— **IV** ———

**Appendices**



# A

## Developing a Customized Application

Customization Overview .....	92
Entity Overview .....	92
Localizing Labels and Fields in Your Application .....	92
Adding Saved Filters to Your Mashup.....	93
Using PTC Creo View Extension .....	94
Using Info*Engine Tasks when Developing Your Application .....	95
Querying WTParts By Name .....	96
Using Custom Reports from Windchill .....	100
Using Windchill REST Web Services when Developing Your Application .....	102

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

# Customization Overview

This chapter describes advanced customization capabilities and is intended for PTC Windchill system administrators and application developers who are using ThingWorx Composer to build ThingWorx applications that connect to Windchill.

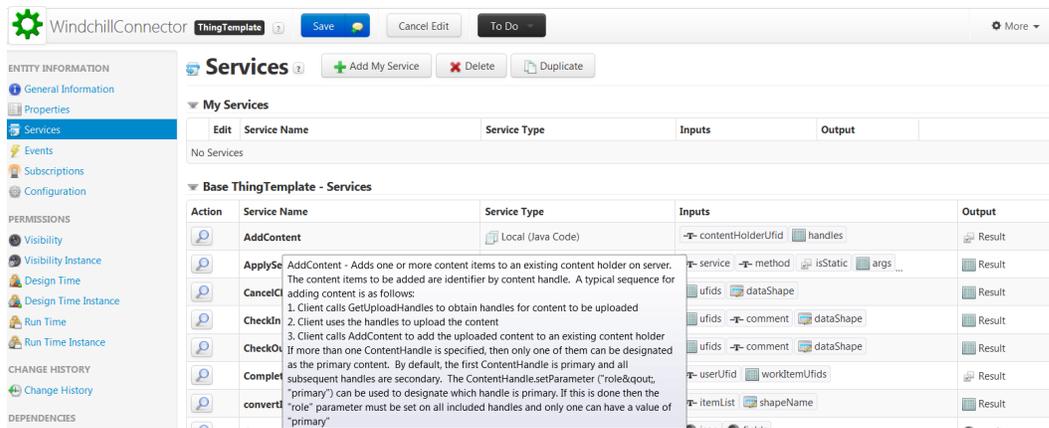
Customization assumes the following capabilities:

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

## Entity Overview

The PTC Windchill Extension includes an entity called the **WindchillConnector**. This entity is a thing template representing an interface to a Windchill system. It includes services that provide the means to access Windchill data and the processing power of the Windchill system.

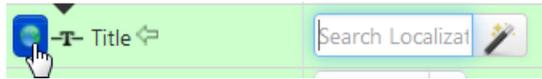
After installation and configuration is complete, learn more about the **WindchillConnector** thing template by opening ThingWorx Composer and viewing its list of services. As shown in the screen below, you can hover over a service to view additional information about the service.



## 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, you can choose to 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:

- The **label** field is the value shown in the drop-down widget.
- The **value** field 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.

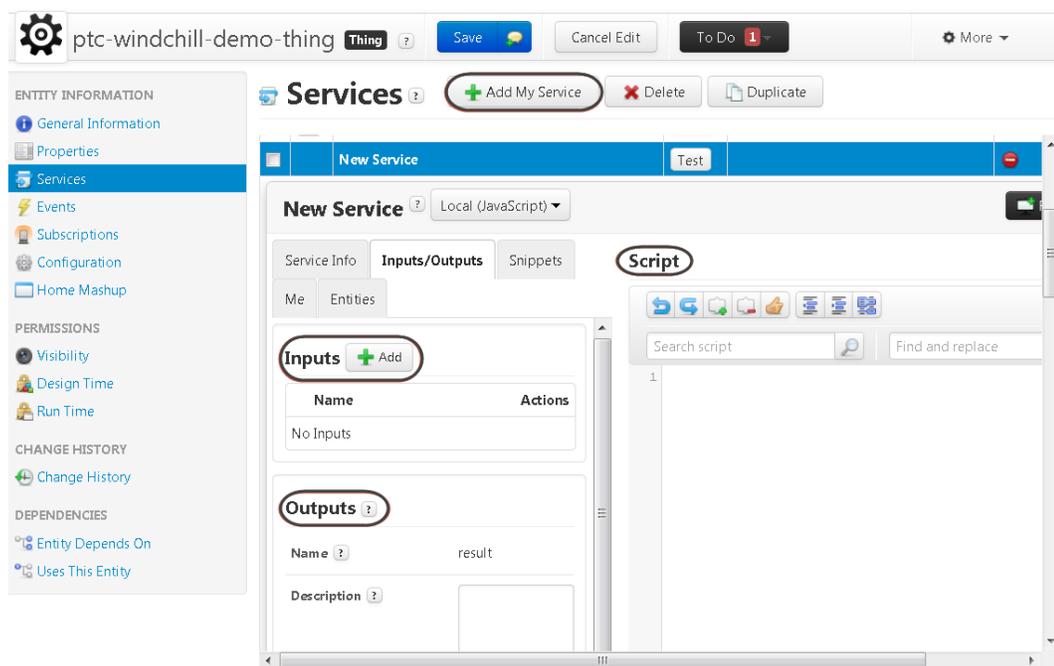
- OccurrenceField – This property enables you to select the column that must be used as the Occurrence ID. The column is selected from the data table which is bound to the Data property.
- selectedOccurrencePath – This property specifies the Occurrence ID of the part selected in the 3D data.
- selectionChanged – 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, [Querying WTParts By Name on page 96](#) and [Using Custom Reports from Windchill on page 100](#) describe some common application features that you can include using Info\*Engine tasks.

## 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 and are implementing a new service from within the thing that will be 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:webobject 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:webobject>
```

---

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:**

STRING name

### **Input Parameter**

Name ?	<input type="text" value="name"/>
Description ?	<input type="text"/>
Base Type ?	<input type="button" value="-T- STRING ▼"/>
Required ?	<input checked="" type="checkbox"/>
Has Default Value ?	<input type="checkbox"/>

- **Outputs:**

**Base Type:** INFOTABLE

**Data Shape:** my.wt.part.WTPart

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

**Outputs** ?

<b>Name</b> ?	result
<b>Description</b> ?	Query Part results
<b>Base Type</b> ?	INFOTABLE ▼
<b>Data Shape:</b> ?	wt.part.WTPart ✎
<b>Infotable Type</b> ?	Just Infotable ▼

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

---

 **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.

---

```
/*
 * 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](http://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 will do:

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, you must create the Data Shapes that this service needs. Specifically, you need to complete the following steps:

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

The report template object contains BLOBed XML data which cannot be fetched by Info\*Engine. 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 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`.

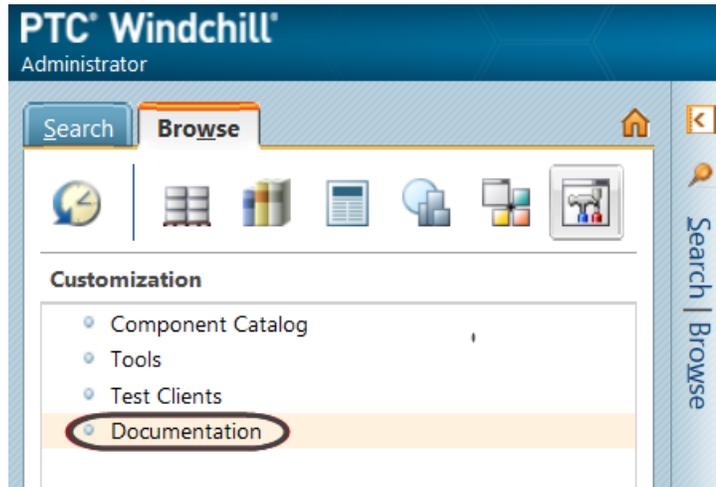
# Using Windchill REST Web Services when Developing Your Application

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**:

Preference Management Standard

Name	Value	Description
Add to Baseline		Add to Baseline operation preferences
Advanced Assembly Editor		Advanced Assembly Editor Preferences
Arbortext		Arbortext Preferences
Associativity		Contains the preferences that are specific to the associativity area.
Attachments		
Attribute Handling		
Business Administrative Change Promotion		
Business Rules		
Change Management		Change Management preferences
Client Customization	Yes	Value that determines if the client customization examples and tools should be enabled in the UI
Configurable Link		Configurable Link preference

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

## ServiceTimeout Property

The **ServiceTimeout** property controls the default length of 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 makes a call to 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) + "/result",
  timeout: 10000
};
var jsonResult = me.processJSONRequest(params);

```

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

The screenshot shows the configuration page for 'WindchillConnectorThing'. The left sidebar contains navigation options under 'ENTITY INFORMATION', 'PERMISSIONS', and 'CHANGE HISTORY'. The main content area is titled 'Configuration for WindchillConnectorThing' and features a table of 'PTC Connection Settings'.

Name	Value
ServiceTimeout	10000
searchParams	Search DataShapes or + to create a new one
baseURL	http://localhost:7080/Windchill
navigationParams	Search DataShapes or + to create a new one
restPath	/servlet/rest

# B

## 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.
    - In Windchill 11.0 and 10.2, the most recent version of the PTC HTTP Server is included with the CPS.
    - In Windchill 10.1, the Early Release Download is available from PTC Technical Support at [http://support.ptc.com/appserver/wcms/standards/freefull.jsp?im\\_dbkey=73856&icg\\_dbkey=893](http://support.ptc.com/appserver/wcms/standards/freefull.jsp?im_dbkey=73856&icg_dbkey=893).
- 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. Then follow these steps to configure SSL. If you choose to install the PTC HTTP Server with the option to **preserve** configuration (the default option), then you need to merge the configuration scripts before you can use these steps to configure SSL.
- 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.
  - Steps 4, 6, and 8 refer to files available in Windchill 10.2 and earlier.
  - 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. Create a truststore for ThingWorx and import the Windchill certificate:

```
keytool -import -alias <some alias name> -file <path to certificateAuthority.cert>
        -storetype jks -keystore <path to new truststore>
```

### 2. Create keystore for ThingWorx.

#### a. Create a self-signed certificate:

```
openssl req -nodes -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days
        <days to certify certificate>
```

#### b. From the certificate, create a p12 format key and be sure to provide an export password:

```
openssl pkcs12 -export -name servercert -in selfsignedcert.crt
        -inkey serverprivatekey.key -out myp12keystore.p12
```

#### c. Import this certificate into a new key store:

```
keytool -importkeystore -destkeystore <path to new keystore> -srckeystore
        myp12keystore.p12 -srcstoretype pkcs12 -alias <some alias name>
```

- 
3. Configure Windchill to trust ThingWorx.
    - 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. Make a note of 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 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. Add a new line `SSLOptions +ExportCertData`.
    - 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]
-Dresource=sslClientAuth -DresourceAuthType=sslClientAuth addAuthResource
```
    - iii. Configure Windchill to trust the ThingWorx certificate.
      - 1) Edit `<windchill-home>/codebase/WEB-INF/web.xml`.

- 
- 2) Find `<filter-name>TrustedSSLAuthFilter</filter-name>`.
  - 3) 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>
```

- 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 no configuration errors exist.
  5. Launch **ptc-connected-demo-mashup** to verify the ThingWorx – Windchill trusted SSL configuration.

# C

## Architecture of Single Sign-On

Design of Authentication Filter .....	110
Design of Authenticator Extension .....	111

You can use the PTC Identity Provider (IdP) 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.

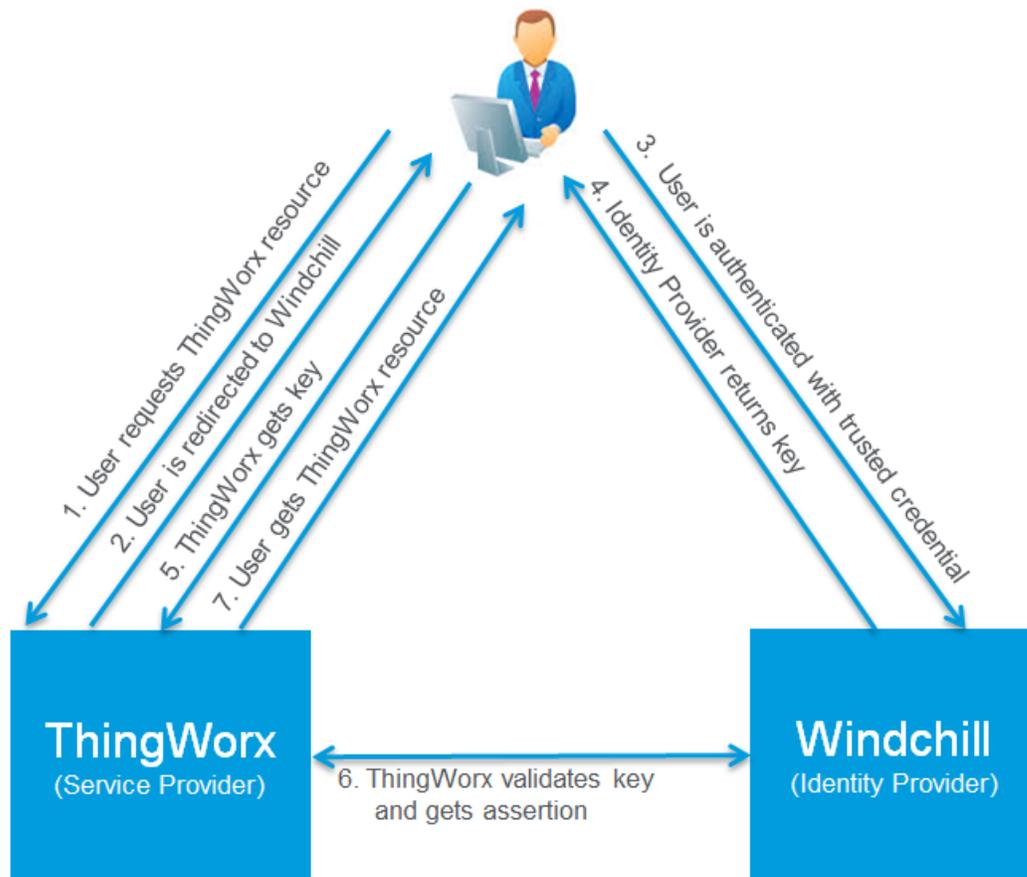
The architecture consists of two components:

- **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 get access to 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 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 [Add Servlet Filters to Tomcat Configuration on page 41](#).

## 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 39](#).

- 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 39](#).