



ptc navigate™

## **Installation and Configuration Guide**

**PTC Navigate Framework Extension 1.2.1**  
**PTC Navigate View PLM App Extension 1.3**  
**PTC Navigate View ALM App Extension 1.1**

---

## **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

About the PTC Navigate Installation and Configuration Guide.....	5
PTC Navigate Overview.....	7
Product Requirements for PTC Navigate 1.3.....	8
Release Updates for PTC Navigate.....	9
Installing PTC Navigate.....	12
Upgrade Overview.....	18
Known Issues and Considerations for PTC Navigate.....	21
Changing the Language Setting.....	22
PTC Navigate View PLM App Extension.....	25
PTC Navigate View PLM App Extension Product Overview.....	26
Product Requirements for PTC Navigate View PLM App Extension.....	27
Installing PTC Navigate View PLM App Extension.....	29
Configuring PTC Navigate View PLM App Extension.....	30
Configuring ThingWorx Single Sign-On when Connecting to Windchill.....	35
Tailoring PTC Navigate View PLM App Extension.....	43
PTC Navigate View ALM App Extension.....	67
PTC Navigate View ALM App Extension Product Overview.....	68
Product Requirements for PTC Navigate View ALM App Extension.....	68
Importing the PTC Navigate View ALM App Extension.....	69
Configuring PTC Navigate View ALM App Extension.....	70
Tailoring PTC Navigate View Requirements App.....	73
Known Limitations in PTC Navigate View Requirements App.....	75
PTC Navigate Reference Information.....	79
Advanced Customization.....	80
Example Configuration Using SSL for Secure Communications.....	96
Architecture of Single Sign-On.....	99



# 1

## About the PTC Navigate Installation and Configuration Guide

This guide is the PDF format of the documentation for the following products:

- PTC Navigate Framework Extension 1.2.1
- PTC Navigate View PLM App Extension 1.3
- PTC Navigate View ALM App Extension 1.1

The same documentation set is available as a help center here: [PTC Navigate Help Center](#).



# 2

## PTC Navigate Overview

Product Requirements for PTC Navigate 1.3 .....	8
Release Updates for PTC Navigate .....	9
Installing PTC Navigate .....	12
Upgrade Overview.....	18
Known Issues and Considerations for PTC Navigate.....	21
Changing the Language Setting .....	22

PTC Navigate provides viewing apps that allow you 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 26](#) and [PTC Navigate View ALM App Extension Product Overview on page 68](#).

---

# Product Requirements for PTC Navigate

## 1.3

This topic lists the product requirements for PTC Navigate. For information on the app-related product requirements, see [Product Requirements for PTC Navigate View PLM App Extension on page 27](#) and [Product Requirements for PTC Navigate View ALM App Extension on page 68](#).

### Server Platform Support

- ThingWorx 7.1 only
- PostgreSQL 9.4.x, no less than 9.4.5
- ThingWorx Utilities 7.1
- Tomcat 8.0.33
- Server operating system support
  - Window Server 2012
  - Windows Server 2008 R2 SP1
  - Red Hat Enterprise Linux (RHEL) 7 Update 1

For more information, see [Installing PTC Navigate on page 12](#).

### Client Platform Support

- Java 8
- Browser support
  - Internet Explorer 11 and later  
PTC Navigate does not support Internet Explorer compatibility mode.
  - Firefox 35 and later
  - Chrome 44 and later
  - Safari 6.1.6 and later

---

#### Note

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

---

Additional information can be found in the [ThingWorx 7.1 System Requirements](#).

---

# Release Updates for PTC Navigate

Here are the highlights of PTC Navigate 1.3.

---

## Note

You must be logged in as an administrator to tailor apps.

---

- Tailor apps to display a link to the object in Windchill. For more information, see [Common Tailoring for Apps on page 44](#).



- Tailor part apps to allow searches on effectivity date. For more information, see [Common Tailoring for Apps on page 44](#).
- Tailor the **View Parts List** app to include design files and select which design file relationships are applicable. For more information, see [Tailoring for Specific Apps on page 49](#).

## ptc navigate™ View Parts List Tailoring

Select role to tailor

---

Data shape for parts list

Include design files link?  Yes

Display which design file formats?

Display which design file relationships?

Select Items

- Described By
- Reference Documents
- Representation
- Attachment

- Tailor the **View Design Files** apps to select which design file relationships are applicable. For more information, see [Tailoring for Specific Apps](#) on page 49.

## ptc navigate™ View Design Files Tailoring

Select role to tailor Administrators

---

Display which formats?

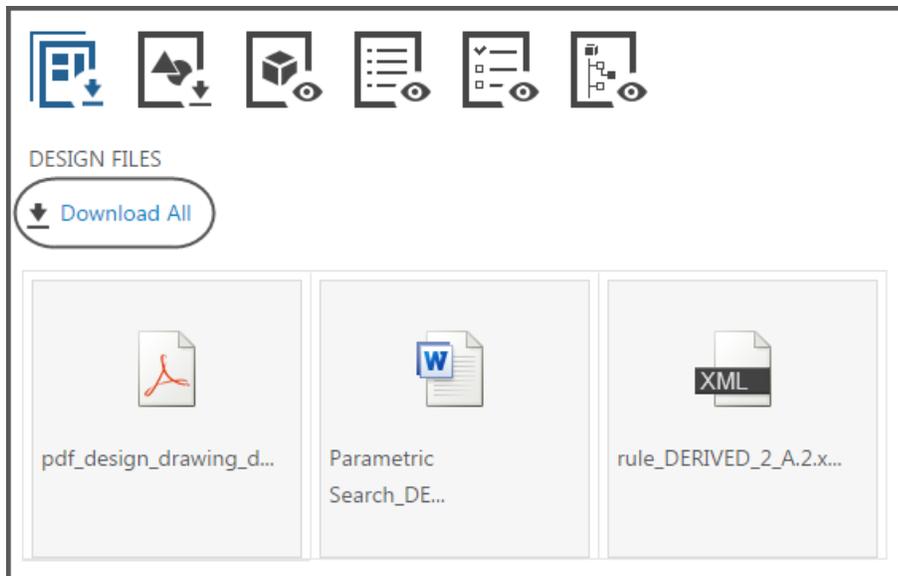
Display which design file relationships?

Select Items

- Described By
- Reference Documents
- Representation
- Attachment

Save Cancel

- In the **View Design Files** app, download all related design files for a part or CAD document in a single ZIP file.



- 
- Lifecycle states available for the **Latest** filter are automatically loaded from Windchill. In previous releases, sites had to update ThingWorx Composer with lifecycle states that were not included by default.
  - In the previous release, **Saved Filter** resolved the part structure but did not affect the search results. Now **Saved Filter** resolves the part structure and filters the search results. For more information, see [Common Tailoring for Apps on page 44](#).
  - Improved performance for the **View Parts List** app when **Include drawings link** is set to **Yes**.
  - Field labels on tailoring pages are updated to enhance readability and usability.

## Installing PTC Navigate

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

### 1. ThingWorx with PostgreSQL

- a. To install PostgreSQL 9.4.x (no less than 9.4.5), go to <https://www.postgresql.org/download/>.
- b. To install and configure ThingWorx, see [Installing ThingWorx 7.1](#).

### 2. ThingWorx Utilities

For more information, see [Importing ThingWorx Utilities Files on page 13](#).

### 3. Windchill Extension

For sites installing the PTC Navigate View PLM App Extension. If your site is installing only PTC Navigate View ALM App Extension, skip this step.

Import the extension files as well as patches specific to the version of Windchill you have installed. For more information, see [Importing Windchill Extension Files on page 15](#).

### 4. PTC Navigate Framework Extension

For more information, see [Importing PTC Navigate Framework Extension Files on page 17](#).

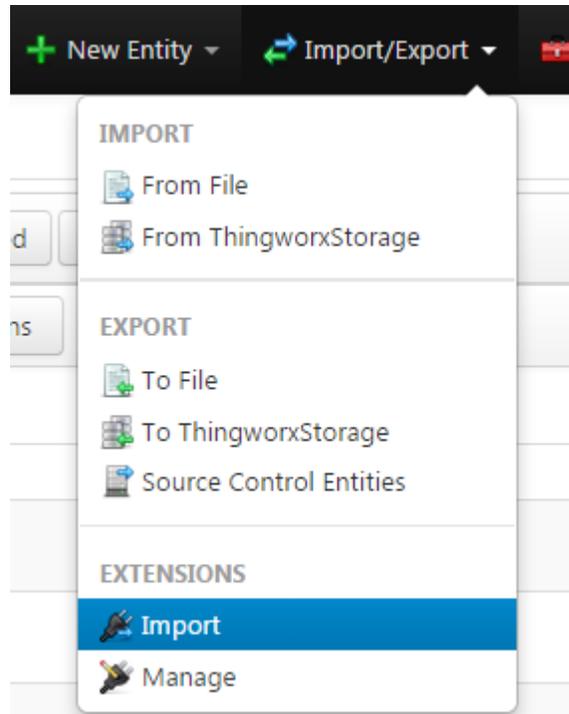
### 5. Product app extensions

- PTC Navigate View PLM App Extension  
For more information, see [Installing PTC Navigate View PLM App Extension on page 29](#).
- PTC Navigate View ALM App Extension

For more information, see [Importing the PTC Navigate View ALM App Extension](#) on page 69.

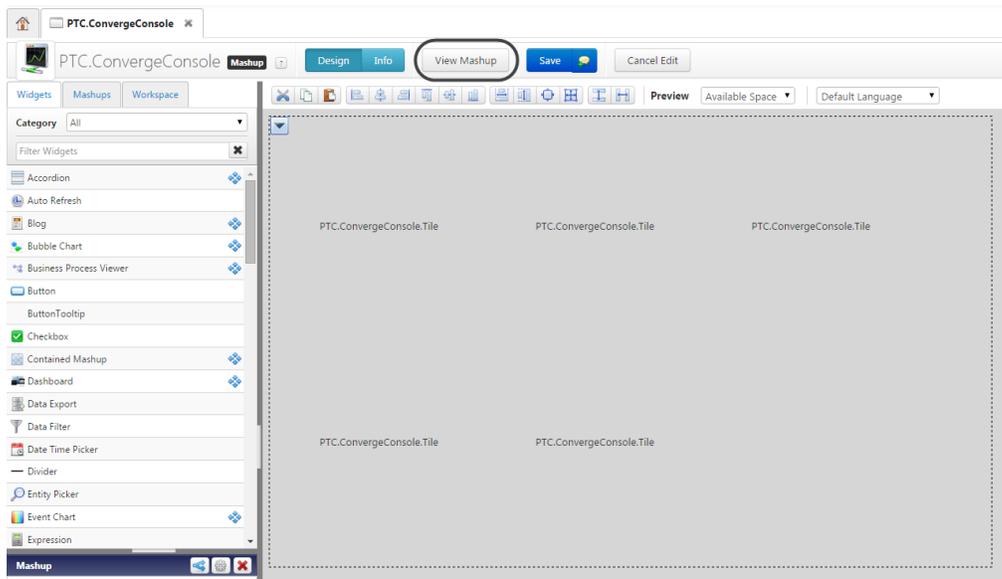
## Importing ThingWorx Utilities Files

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



2. The **Import Extensions** window appears.
3. Navigate to `ThingWorx-Utilities-{version}.zip`, and click **Import**.

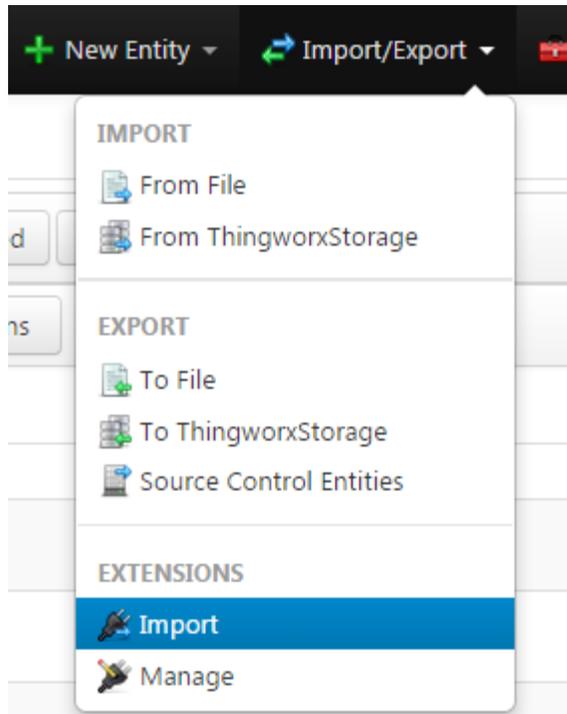
4. Click **Close**. When prompted to refresh your browser, click **Yes**.
5. Verify that the extension has been successfully imported by searching for the **PTC.ConvergeConsole** mashup type. Select the mashup by clicking the name or the view icon. From the Mashup Editor, click **View Mashup** and confirm the mashup displays.



## Importing Windchill Extension Files

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

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



2. In the **Import Extensions** window, click **Browse** and navigate to `ptc-windchill-extension-{version}.zip`.
3. Click **Import**.
4. If prompted, refresh the composer.
5. Select **Import/Export** ► **From File**.
6. In the **Import From File** window, browse to `ptc-sample-mashup-{version}.xml` and click **Import**.

### 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`.

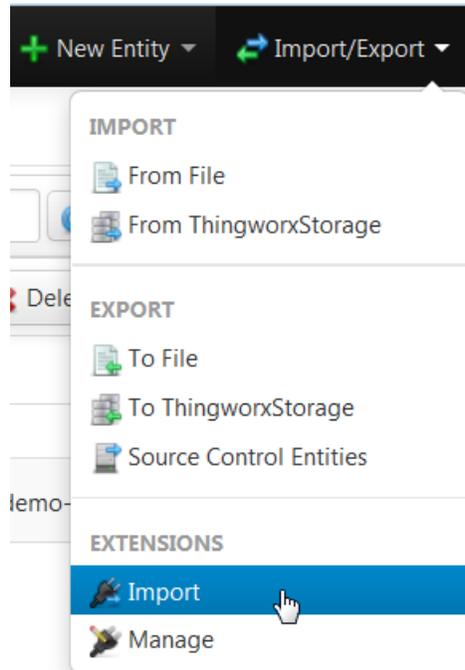
Each patch includes instructions specific to the release. To install the patch, unzip the file and use the instructions found in the `Readme` file included with the patch.



## 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.

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

5. Install the app extensions your site will use.
  - [Installing PTC Navigate View PLM App Extension on page 29](#)
  - [Importing the PTC Navigate View ALM App Extension on page 69](#)

---

## Upgrade Overview

The following considerations apply for all upgrades to PTC Navigate 1.3:

- The upgrade process migrates tailoring options and permissions as well as preserves settings that enable single sign-on with Windchill.
- In PTC Navigate 1.3, ensure the **Display Windchill navigation link** field on the [common tailoring page on page 44](#) is set to **Yes**, the default value. Changing the value can cause upgrade issues.
- Lifecycle states from Windchill are loaded in PTC Navigate 1.3. Go to the [common tailoring page on page 44](#) to select lifecycle states. Previous selections are not transferred with the upgrade.
- Recent items are not carried forward to the new release. Only items viewed since installing the new release display under the **RECENT** tab.

Proceed to the upgrade instructions for your version:

- [Upgrading PTC Navigate from 1.2.x to 1.3 on page 18](#)
- [Upgrading PTC Navigate from 1.1 to 1.3 on page 19](#)

## Upgrading PTC Navigate from 1.2.x to 1.3

1. Install a new instance of ThingWorx 7.1 on a separate server.
2. Export the `AllEntities` file from 1.2.
  - a. In ThingWorx Composer, select **Import/Export** ► **Export** ► **To File**.  
The **Export to File** window opens.
  - b. Under the **Entities** tab, confirm **Binary** is selected in the **ExportType** field, and click **Export**.  
This exports all entities into the file `AllEntities.twx`.
3. Import the `AllEntities` file in 1.3.
  - a. In ThingWorx Composer, select **Import/Export** ► **Import** ► **From File**.  
The **Import From File** window opens.
  - b. Select the **Entities** button.
  - c. Under the **Single File** tab, browse to `AllEntities.twx` and click **Import**.

---

 **Tip**

If your site created custom groups in PTC Navigate and added the **PLMAppsRolesTag** as described in [Modify ThingWorx Permissions on page 39](#), that tag is retained on upgrade.

---

## Upgrading PTC Navigate from 1.1 to 1.3

### Prerequisites

- Upgrading from PTC Navigate 1.1 to 1.3 requires upgrading from ThingWorx 6.5 to 7.1. For more information, see [Upgrading to ThingWorx 7.1](#).

---

 **Note**

Install ThingWorx 7.1 in a separate location from ThingWorx 6.5 (using a new instance of Tomcat for ThingWorx 7.1).

---

- 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.
- PTC Navigate 1.1 entities cannot have permissions set to the users role. In ThingWorx, go to the following entities and remove any permissions set for the users role.
  - Example1DesignFilesAccessAppConfigThing
  - Example1DrawingAccessAppConfigThing
  - Example1InterrogateCadModelAccessAppConfigThing
  - Example1PartPropertiesAccessAppConfigThing
  - Example1PartsListAccessAppConfigThing
  - Example1PartStructureAccessAppConfigThing
  - Example1DocumentAccessAppConfigThing
  - AppCollectionsConsolidator
  - MasterAccessAppThing
  - PTC.Converge.App.BaseAccessAppTemplate
  - PTC.Converge.App.RecentlyViewedItemsTemplate

- PTC.Resource.ResourceProviderTemplate

## A. Backup Configuration Data

Before removing a previous release of PTC Navigate, backup your configuration using ThingWorx import and export functionality.

Use the following procedure to export and edit the `AllEntities.xml` file.

1. In ThingWorx Composer, select **Import/Export** ► **Export** ► **To File**.  
The **Export to File** window opens.
2. Under the **Entities** tab, change the **ExportType** field to **Xml** and click **Export**.  
This exports all entities into the file `AllEntities.xml`.
3. Edit the `AllEntities.xml` file to update the `ptc-creo-view-extension` package version. The version format used in 1.1 for this extension is no longer valid.
  - a. Search for the string `ptc-creo-view-extension`.
  - b. Change the value of the `packageVersion` attribute from `10.2.30.38` to `10.2.30` or any string that matches the major.minor.patch format.
  - c. Save the `AllEntities.xml`.

## B. Install PTC Navigate

Install the extensions for PTC Navigate in the required order.

1. [Importing ThingWorx Utilities Files on page 13](#)
2. [Importing Windchill Extension Files on page 15](#)
3. [Importing PTC Navigate Framework Extension Files on page 17](#)
4. [Installing PTC Navigate View PLM App Extension on page 29](#)
5. [Importing the PTC Navigate View ALM App Extension on page 69](#)

## C. Restore Configuration Data

After installing the new release, restore the configuration data you backed up previously. Use the following procedure to import the `AllEntities.xml` file.

1. In ThingWorx Composer, select **Import/Export** ► **Import** ► **From File**.  
The **Import From File** window opens.
2. Select the **Entities** button.
3. Under the **Single File** tab, browse to your `AllEntities.xml` file and click **Import**.

---

## D. Post-Upgrade Steps and Considerations

- The upgrade process migrates tailoring options as well as user group and member permissions. If your site created custom groups in PTC Navigate 1.1, follow the instructions in [Modify ThingWorx Permissions on page 39](#) to add the **PLMAppsRolesTag**.
- If your site configured single sign-on with Windchill for PTC Navigate 1.1, the upgrade preserves the settings for **ptc-identity-provider-authenticator**. After the upgrade, servlet filters need to be added to Tomcat. Follow the steps in [Add Servlet Filters to Tomcat Configuration on page 37](#).
- In ThingWorx Composer, verify the configuration data was migrated correctly in **ptc-windchill-demo-thing**.

## Known Issues and Considerations for PTC Navigate

The following items describe circumstances that could exist in PTC Navigate and PTC Navigate View PLM App Extension. For information specific to the PTC Navigate View ALM App Extension, see [Known Limitations in PTC Navigate View Requirements App on page 75](#).

### Apostrophe in Searches

- Issue: Using an apostrophe (') in an app search, returns an error.
- Workaround: Add an additional apostrophe. For example, instead of `piston's bolt`, enter `piston''s bolt`.
- Resolution: This issue is resolved in a future release of PTC Navigate.

### View Design Files

The behavior of the **View Design Files** app varies among browsers. For example, one browser asks whether to open or save files while another browser opens some files automatically.

### Users and Groups

- Issue: Once a user is added to a group, it cannot be deleted from the group, although the action is available in ThingWorx Composer and the user appears deleted.
- Workaround: Ensure each user is correctly added to a group before doing role-based tailoring on [common on page 44](#) and [app-specific on page 49](#) tailoring pages.
- Resolution: This issue is resolved in a future release of ThingWorx.

---

## Form-Based Authentication

- Issue: PTC Navigate is not working when the Windchill 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.

## Changing the Language Setting

Use the following procedure to change your language setting for PTC Navigate.

1. In ThingWorx Composer, go to **SECURITY ► Users**.
2. Select and edit the user, in this case Administrator.

The screenshot shows the user configuration interface for the 'Administrator' user. The top bar includes a 'Save' button, a 'Cancel Edit' button, and a 'To Do' button. The sidebar on the left is titled 'ENTITY INFORMATION' and contains the following sections:

- General Information** (selected)
- User Extensions
- User Profile Configuration
- PERMISSIONS**
  - Visibility
  - Design Time
  - Run Time
- CHANGE HISTORY**
  - Change History
- DEPENDENCIES**
  - Entity Depends On
  - Uses This Entity

The main panel is titled 'General Information' and contains the following fields:

Name	Administrator
Description	Administrator
Project	Search Projects or + to create a new
Tags	Search ModelTags or + to create a r
Languages	en <span>Edit</span>
Password	<span>Change Password</span>
Enabled	<input checked="" type="checkbox"/>

3. In the **Languages** field:

- Click **Edit** to open a window in which you can drag-and-drop the language preferences.
- or
- Enter the abbreviation for the language you prefer, for example, `en` for English or `ko` for Korean. Abbreviations are listed in the **Edit** window.



# 3

## PTC Navigate View PLM App Extension

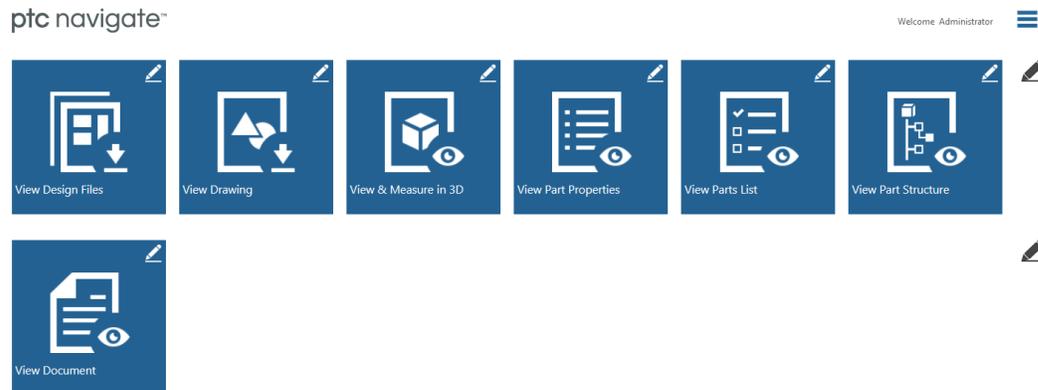
PTC Navigate View PLM App Extension Product Overview .....	26
Product Requirements for PTC Navigate View PLM App Extension .....	27
Installing PTC Navigate View PLM App Extension .....	29
Configuring PTC Navigate View PLM App Extension.....	30
Configuring ThingWorx Single Sign-On when Connecting to Windchill.....	35
Tailoring PTC Navigate View PLM App Extension .....	43

The apps available with the PTC Navigate View PLM App Extension provide visibility to parts, documents, and CAD document stored in Windchill. The topics in this section describe how to install and configure the extension and tailor the apps.

---

# PTC Navigate View PLM App Extension Product Overview

PTC Navigate View PLM App Extension provides viewing apps that allow you 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 [Common Tailoring for Apps on page 44](#) and [Tailoring for Specific Apps on page 49](#).

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
```

---

## **i** Best Practice

- PTC Navigate apps were designed for a screen resolution of 1280x1024. Results on other resolutions vary.
- App searches could require extra time when calling Windchill data. For more information, see [Editing Thing Configuration on page 42](#).

---

## App Collections

Apps are grouped into related collections, one for parts and one for documents. The app collections provide the following benefits:

- You 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.

#### PART DETAILS



**Number** GC000001  
**Name** GOLF\_CART  
**Version** A.1 (Manufacturing)  
**State** In Work

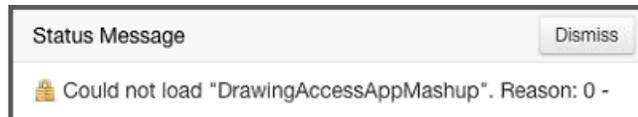
[View in Windchill](#)



- 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 [Common Tailoring for Apps on page 44](#).

#### 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.



## Product Requirements for PTC Navigate View PLM App Extension

PTC Navigate View PLM App Extension requires the following product support.

- Windchill at one of the following release levels:
  - 11.0 M010
  - 11.0 F000
  - 10.2 M030 CPS 01 or later
  - 10.2 M020 CPS 06 or later

---

**Note**

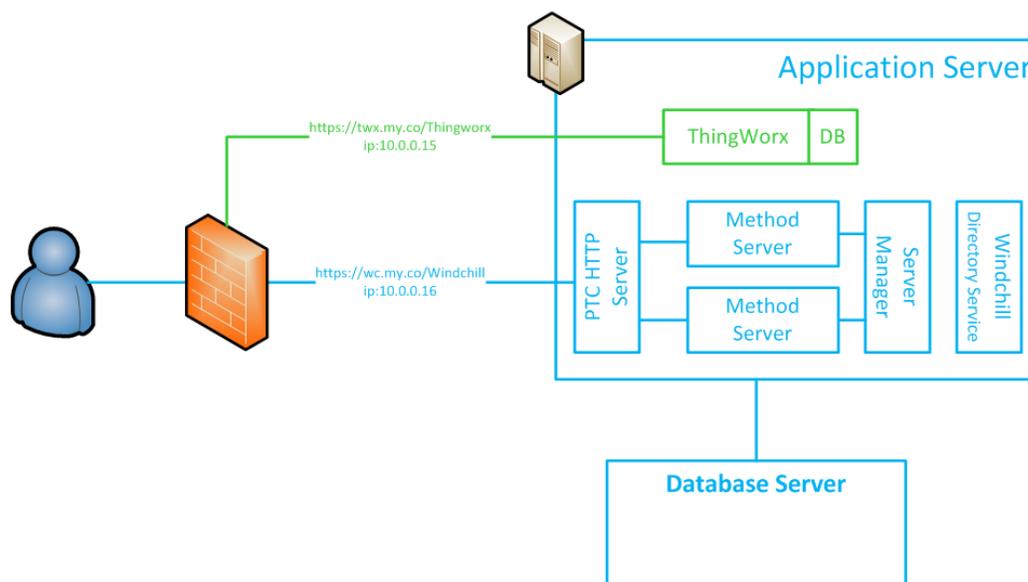
Windchill system requirements information for your specific Windchill release is available on the [Reference Documents](#) page.

---

- Windchill Extension 1.5, which includes a patch that corresponds with the Windchill release you have installed.
- 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 30](#). The required Windchill security fixes are available from PTC:
  - For Windchill 10.2 M020, update to CPS06 or later for required security fixes.
  - For Windchill 10.2 M030 and later releases, the PTC HTTP Server bundled with the release has the required security fixes for using SSL.

PTC recommends that you install the CPS or Early Release Download with the option to **overwrite** the PTC HTTP Server configuration and redo the configuration. If you install the PTC HTTP Server with the option to **preserve** the configuration (the default option), merge the configuration scripts before you configure SSL. For more information, see [Example Configuration Using SSL for Secure Communications on page 96](#).

The following diagram depicts a ThingWorx deployment with Windchill.



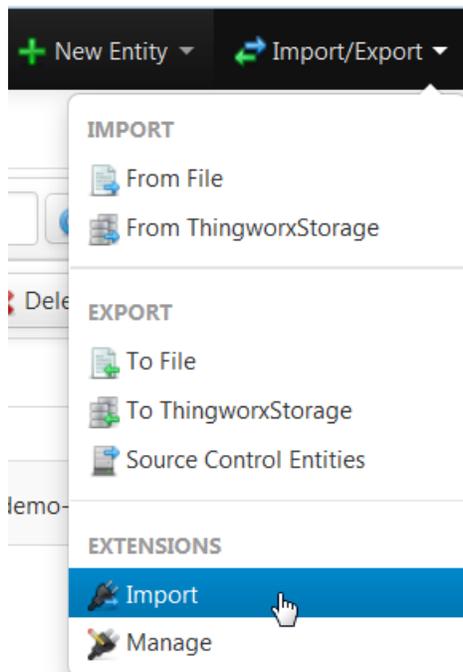
# Installing PTC Navigate View PLM App Extension

## Prerequisites

Install prerequisite products described in [Installing PTC Navigate on page 12](#) before installing the PTC Navigate View PLM App Extension.

## Importing PTC Navigate View PLM App Extension Files

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. Browse to PTCNavigate\_Example-{version}-imports.xml, and click **Import**.

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

---

# Configuring PTC Navigate View PLM App Extension

This section contains configuration information for ThingWorx and Windchill systems.

Systems using the PTC Navigate View ALM App Extension could 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 communication is configured in the next step as part of using SSL to establish a trusted relationship.
- Trust between Windchill and ThingWorx that is based on SSL public/private keys. Follow the steps in [Using SSL for Secure Communication on page 31](#).
- 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 35](#).

---

## 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 34](#).
- 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 Permissions on page 39](#). Also update the `web.xml` file as described in [Add Servlet Filters to Tomcat Configuration on page 37](#).

## 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 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 35](#).

## Using SSL for Secure Communication

Your site can use two-way SSL to secure the communication between the ThingWorx server and the Windchill application. PTC recommends using SSL when working in a production environment. For development and demo environments, see [Using Trusted Host on page 34](#). 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 96](#).

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

---

 **Note**

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

---

## Configuring ThingWorx

ThingWorx requires Java key stores and trust stores.

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

SSL Connection Configuration	
Name	Value
SSLKeyStoreType	<input type="text" value="JKS"/>
SSLKeyStorePath	<input type="text" value="/usr/local/tomcat/twxjks"/>
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.
  - For Windchill 10.2 releases:
    - a. Navigate to and edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.
    - b. Uncomment the `SSLCACertificateFile` entry so that it refers to the `ca-bundle.crt` file. Make a note of the location of the `ca-bundle.crt` file.
    - c. Save changes.
  - For Windchill 11.0 releases:
    - a. Create a file in `<APACHE_HOME>\conf\ca-bundle.crt`. This location is recommended but not required.
    - b. Paste the contents of `Server.crt` file used in HTTP Server.
    - c. Save `ca-bundle.crt`.
    - d. Create a `sslclientauth.conf` file at `<APACHE_HOME>\conf\sslvhostconf.d`.
    - e. Add `SSLCACertificateFile <PATH_TO>\ca-bundle.crt` in it so that it refers to the `ca-bundle.crt` file.
    - f. Save changes.
2. Add the ThingWorx certificate to PTC HTTP Server's list of trusted CA certificates.
  - a. Edit the `ca-bundle.crt` file you pointed to previously in the `SSLCACertificateFile` entry.

---

### Note

If the file does not yet exist, create the file and any required directories.

---

- b. Append the contents of the ThingWorx certificate held in the trust store to this file.
  - c. Save changes.
3. Configure PTC HTTP Server for the sslClientAuth URL.
  - a. Find the Windchill web app name in `<windchill-home>/codebase/wt.properties` in the entry `wt.webapp.name`.
  - b. Open a shell or command prompt and enter the following:
 

```
cd <HTTPSERVER_HOME>ant -f webAppConfig.xml -DappName=[windchill-web-app]
-Dresource=sslClientAuth -DresourceAuthType=sslClientAuth addAuthResource
```
  - c. Configure Windchill to trust the ThingWorx certificate.
    - i. Edit `<windchill-home>/codebase/WEB-INF/web.xml`.
    - ii. Find `<filter-name>TrustedSSLAuthFilter</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 31](#).

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 99](#).

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

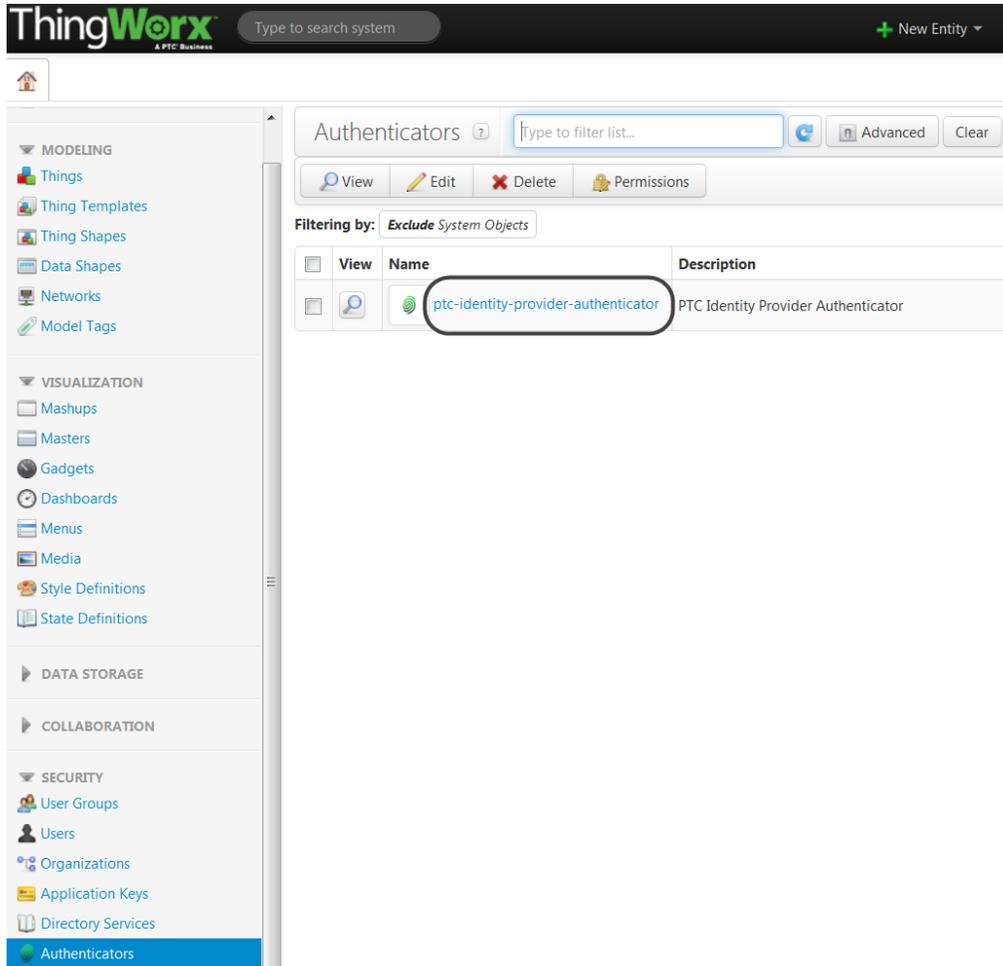
Use the steps in the following sections to configure single sign-on using Windchill as the identity provider:

1. [Configuring the PTC Identity Provider Authenticator on page 36](#)
2. [Add Servlet Filters to Tomcat Configuration on page 37](#)
3. [Modify ThingWorx Permissions on page 39](#)
4. [Verifying Single Sign On on page 42](#)

## 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.

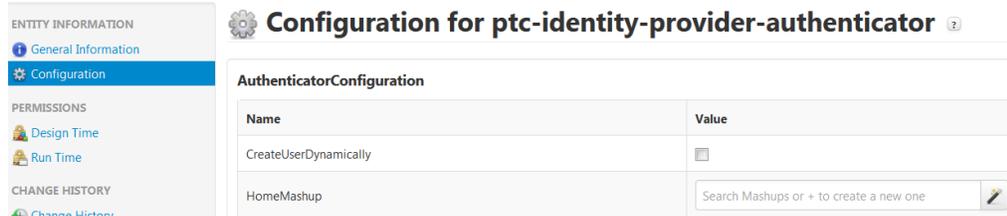


The **General Information** page opens.

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.

- Click **Configuration** under **ENTITY INFORMATION** in the left navigation pane:



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

- **CreateUserDynamically** – When selected, configures the authenticator to automatically create a user in ThingWorx if the user does not yet exist and the user has been authenticated by Windchill. If the authenticator is not configured to automatically create users, the browser is still routed to Windchill for authentication but then fails to open ThingWorx if the users does not exist.
- **HomeMashup** – Assigns a home mashup to the newly created users. If no home mashup is specified, ThingWorx routes the browser to the ThingWorx search page.

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

- 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.
- 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 35](#).

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

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

---

 **Note**

The JAR file is included with the extension files in `Framework.zip`.

---

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

---

 **Note**

Note the following about code examples:

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

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

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

```

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

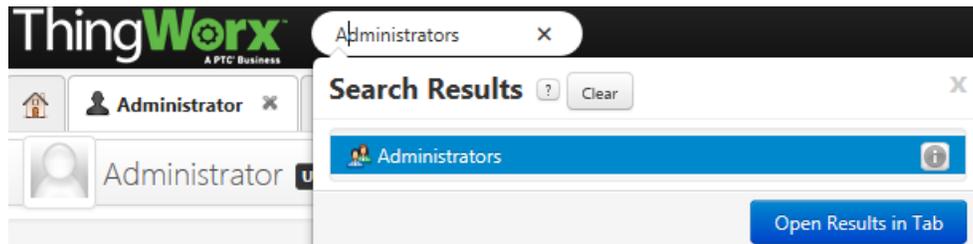
```

5. Restart Tomcat.

## Modify ThingWorx Permissions

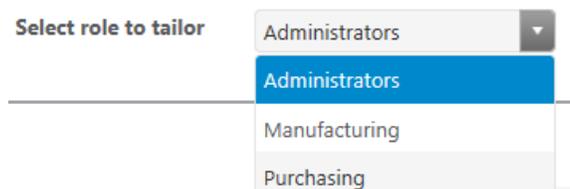
### Administrators

To be a PTC Navigate administrator, you must 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 Composer screen.



### Users and Groups

In ThingWorx Composer, create user groups for each role your site wants to include in PTC Navigate. The groups enable role-based tailoring and display in the **Select role to tailor** list on [common on page 44](#) and [app-specific on page 49](#) tailoring pages.



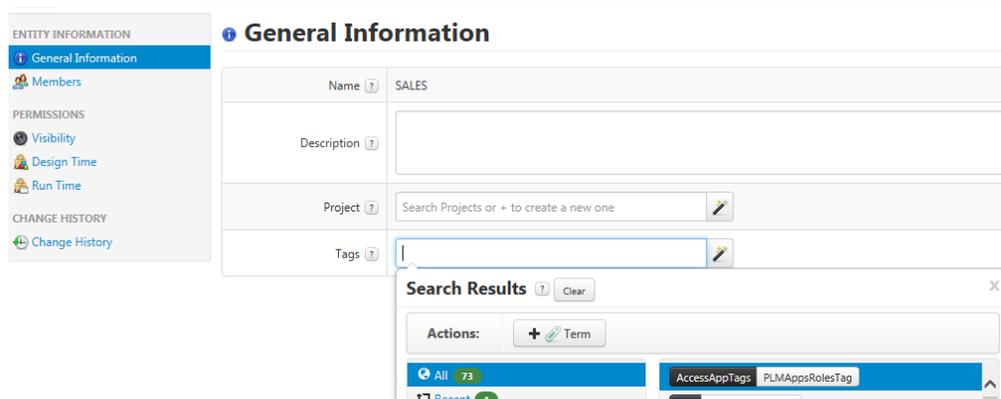
1. In ThingWorx Composer, go to **SECURITY ► User Groups**.
2. Create new groups as desired at your site.
3. Manufacturing and Purchasing groups are included with PTC Navigate by default.

Proceed to the step to create and add users for these groups, or delete these groups if inapplicable at your site.

### Note

If your site upgraded from PTC Navigate 1.1 and had custom groups named Manufacturing or Purchasing, add the tag described in the next step.

4. For new groups to display as roles on tailoring pages, edit each group to add the following tag: **PLMAppsRolesTag** under **AccessAppTags**.



### Note

If the `PLMAppsRolesTag` tag is removed from any group, the role does not display on [common on page 44](#) and [app-specific on page 49](#) tailoring pages.

5. Create and add users as members in each group. Note these important considerations:
  - Users must be assigned to a group to see any PTC Navigate apps.
  - If a group is deleted, users assigned to the group see a blank page without any PTC Navigate apps.

- 
- If you determine that a group is no longer relevant for PTC Navigate, complete the following steps:
    - a. In PTC Navigate, use the options on the [common on page 44](#) tailoring pages to hide all apps from the group.
    - b. In ThingWorx Composer, remove the `PLMAppsRolesTag` tag from the group.

---

 **Note**

If the `PLMAppsRolesTag` tag is removed from any group before you hide the apps on the [common on page 44](#) tailoring pages, the apps remain visible to the users of the group.

---

- Add users to only one ThingWorx user group (other than the generic Users group). Otherwise, the role-based app visibility does not work as expected.
- Once a user is added to a group, it cannot be deleted from the group, although the action is available in ThingWorx Composer and the user appears deleted. This is a known issue in ThingWorx 7.1.

---

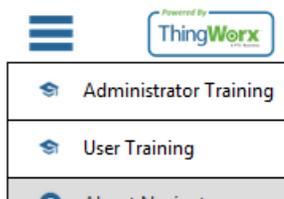
 **Note**

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.

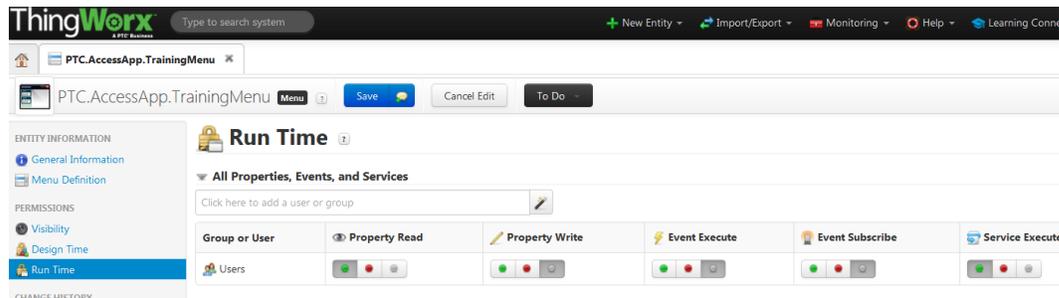
---

## Permissions for Training Menu

The training menu options available in the upper right corner of the PTC Navigate screen are 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.



## 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. For more information, see [Configuring ThingWorx Single Sign-On when Connecting to Windchill on page 35](#).
    - **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 is not used.
  4. Click **Save**.

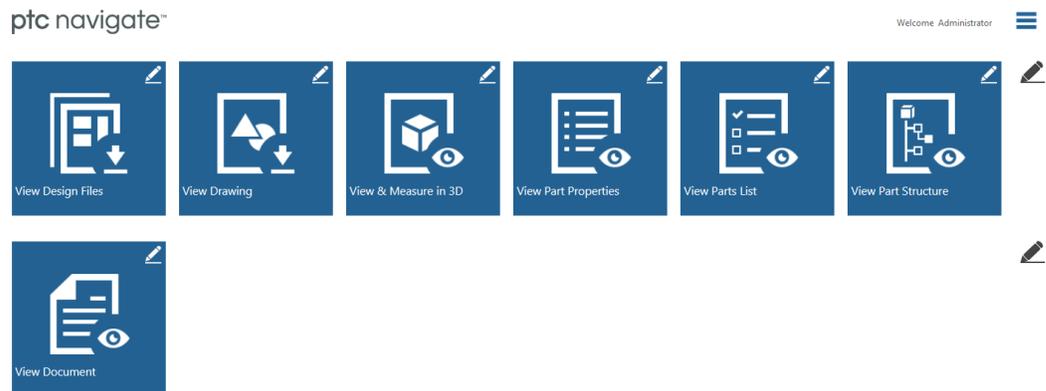
## Tailoring PTC Navigate View PLM App Extension

Administrators can modify the display and function of apps provided with the PTC Navigate View PLM App Extension.

[Common tailoring on page 44](#) pages allow changes across a collection of apps, for example, all part-related or document-related apps. Click the edit collection icon  on the landing page.

Some tailoring options are available for individual apps as described in [Tailoring for Specific Apps on page 49](#). Click the edit icon  on individual apps on the landing page.

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



More advanced tailoring capabilities, such as modifying properties and adding a new life cycle state are described in [Advanced App Tailoring on page 50](#).

---

#### **Note**

To be a PTC Navigate administrator, you must be added to the ThingWorx Administrators group. For more information, see [Modify ThingWorx Permissions on page 39](#).

---

## Common Tailoring for Apps

Common tailoring pages allow changes across a collection of apps, for example, all part-related or document-related apps. When you access the landing page as an administrator, click the edit collection icon  to the right of the app collection on the [landing page on page 43](#).

### Select Role

The **Select role to tailor** list displays all groups defined in ThingWorx Composer. For more information on adding roles, see [Modify ThingWorx Permissions on page 39](#).

Select a role and then modify the tailoring options and app visibility as desired for users in the role.

### Tailoring Options

The options on the common tailoring pages are described below:

- **Max search results:** Maximum number of objects that display in the search **RESULTS** tab.

If the **Search which type** field is set for both parts and CAD documents, the search results limit applies per object type. For example, if you set the limit to

---

10, the search can return 10 parts and 10 CAD documents for a total of 20 objects.

- **Max recent searches:** Maximum number of objects that display in the **RECENT** searches tab.
- **Data shape for recent searches:** Data shape used in ThingWorx Composer that controls the display of the recent searches in the app. For information on editing data shapes, see [Configuring Attributes on Recent Searches on page 57](#).
- **Display Windchill navigation link:** If yes, a **View in Windchill** link displays with the part details.



- **Search which type:** Select whether to search on parts, CAD documents, or both.  
(Not available for document app.)
- **Return partial match results:** Enable search results to return more than exact matches.
- **Search which attribute:** Select whether to search on the object number, name, or both.
- **Use search filter:**

If no, uses the Windchill default saved filter when searching in the apps. Or you can specify a saved filter in the **Default saved filter** field. For more information, see the section below: Best Practices for Saved Filters.

(Not available for document app.)

If yes, the **Which filter** field display from which you can select a filter to be available when searching from the apps.

- **Saved Filter:** Select multiple Windchill filters to display as a drop-down list in the apps. Users can select from the saved filters when performing a search. For more information, see the section below: Best Practices for Saved Filters.  
(Not available for document app.)
- **Saved Search:** Select Windchill saved searches to display as a drop-down list in the apps. Use saved searches available in the Windchill site context.

---

For more information, see the section below: Best Practices for Saved Searches.

- **Latest:** Select lifecycle states and views 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. For more information, see the section below: Best Practices for Saved Searches.
- **Version:** A **Version** field displays in the app search, so searches could be filtered based on an object version, for example A.1. In addition to entering a version numbers, users must enter information in the search field.
- **Effectivity Date:** A calendar and optionally, a view name list display in the app search, so users can search on effectivity date. Note the following:
  - ◆ If you don't specify a view, the app search returns only parts that do not have a view specified.
  - ◆ If a part does not have an effectivity date set, it will not be returned in the search.
  - ◆ Parts that match the effectivity date will be returned in the results. The latest revision is returned if multiple revisions have the same effectivity date.
  - ◆ Searching on effectivity date is available for parts only.

For additional information, see the Windchill help on effectivity. For example, [Selecting a Date Effectivity Configuration Specification](#).

## Best Practices for Saved Filters

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

- In Windchill:
  - To create or modify a filter, go to an object information page. From the **Structure** tab, edit the filter to your desired settings and save the filter. Make note of the name because the exact name is required in PTC Navigate
  - Ensure the filters are set **Share**.
- In PTC Navigate:
  - Configuration specification filters affect search results and resolve part structures. Other filters, such as attribute or spatial filters, resolve part structures but are not applied in search results.
  - Enter the filter name exactly as it appears in Windchill. Preferably copy and paste the name.

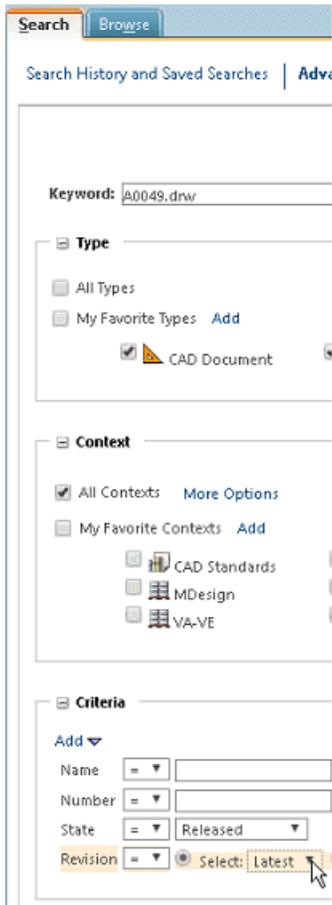
- 
- Enter multiple filter names by separating them with a vertical bar (|) character, no spaces.

### 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, could 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 some tailoring options filter the results of a saved search.
  - If you enter a keyword in a PTC Navigate app search, it overrides the keyword in the saved search.
  - Saved searches work in conjunction with **Search which type** and **Max search results**, but they do not take into consideration the other options on the common tailoring page.
    - ◆ For example, the **View Document** app returns only documents even if the saved search includes additional object types. Or if the **Search which type** field on the tailoring page is set for parts only, CAD documents are not returned, even if CAD documents are included in the saved search.
    - ◆ The **Max search results** tailoring option limits the results of a saved search. Set the limit higher than a saved search query result to ensure that all objects are returned.

As a best practice, create saved searches specific for the object type of each app. A search for multiple object types clutters the **Max search results** and could result in fewer search results of the desired object type.
- 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 following example.



- 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
<b>Latest Version Search</b>	<b>Yes</b>	<b>Search for the latest version of objects.</b>
Recently visited Context Picker list size	5	Recently visited Context Picker list size

---

## Display which apps

Use the toggle buttons above each app icon to determine which apps are available for the role in the **Select role to tailor** field.

---

### Note

The **Display which apps** section does not display when the Administrators role is selected in the **Select role to tailor** field because all apps are available to administrators.

---

## Tailoring for Specific Apps

Options unique to the apps are available on the app-specific tailoring pages. When administrators access the PTC Navigate [landing page on page 43](#), click the edit icon  in the upper right corner of the app.

### Select Role

The **Select role to tailor** list appears at the top of all app-specific tailoring pages. It lists all groups defined in ThingWorx Composer. For more information on adding roles, see [Modify ThingWorx Permissions on page 39](#). Select a role and then modify tailoring options as desired for users in the role.

### Tailoring Options

- **Data shape:** Lists the data shape used in ThingWorx Composer to control the display of the attributes in the app. For information on editing data shapes, see [Configuring Windchill Attributes as App Properties on page 51](#).
- **Include drawings link:** In the **View Parts List** app, include a column that displays a link to the drawing associated with a CAD document. For parts, it returns the drawing associated with the related CAD document.
- **Include design files link:** In the **View Parts List** app, include a column to from which you can download the design file associated with the part. Design files are the described by documents, reference documents, representations, and attachments associated with the object.

In the **Display which design file relationships** field, you select the relationships to include in the search.

- **Described By:** Version-specific relationship of parts and documents. For some product documentation, a version of a part is described by one and only one version of the document.

Not for CAD documents.

- **Reference Documents:** Version-independent relationship of parts and documents. For some product documentation, the latest released version of the document must be connected to the part.  
Not for CAD documents.
  - **Attachment:** Attachments are best used in the following situations:
    - ◆ When content directly correlates to or is derived from the primary content. For example, an image file generated from a technical illustration.
    - ◆ When you do not need to use version control, workflow processes, or customized access control settings for the content.
  - **Representation:** Representations are associated to CAD documents or parts when a CAD document is submitted for publishing.
- **Format fields:** Hint text is available on the PTC Navigate tailoring pages to describe the required information for each app's file format field.

## Advanced App Tailoring

If your site wants to tailor PTC Navigate beyond the options available on the [common on page 44](#) and [app on page 49](#) tailoring pages, administrators need to use ThingWorx Composer to configure the apps. Experience with ThingWorx Composer is recommended. The sections to follow demonstrate some common advanced tailoring options.

Additional customization documentation is available in the reference documentation. See [Advanced Customization on page 80](#).

## App Entity Information

For advanced tailoring configurations, you need the ThingWorx entity information for the apps.

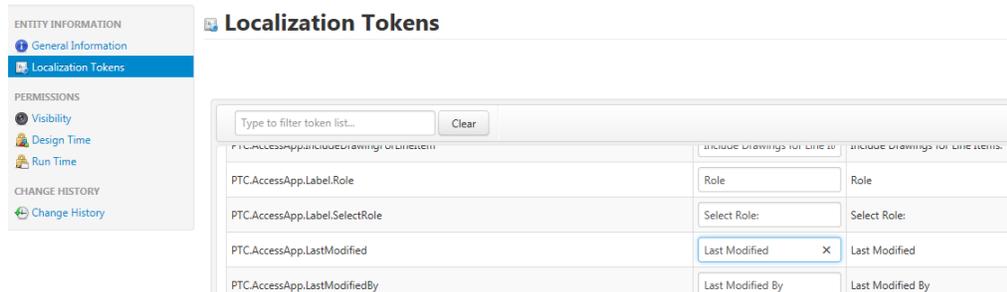
App	Mashup	Data Shape	Data Shape Widget
<b>View Part Properties</b>	PartPropertiesAccessAppMashup	Example1PartPropertiesAccessAppDataShape	PartProperties
<b>View Parts List</b>	PartListAccessAppMashup	Example1PartsListAccessAppDataShape	PartsList

App	Mashup	Data Shape	Data Shape Widget
<b>View Part Structure</b>	PartStructureAccessAppMashup	Ex1PartStructureAccessAppDataShape	PartStructure
<b>View Document</b>	DocumentAccessAppMashup	Example1DocumentAccessAppDataShape	DocumentDetails

## Modifying a Property Label for a Default Attribute

To modify attribute labels included by default with PTC Navigate, use the following procedure. To modify labels for custom attributes created by your site, see [Modifying a Property Label for a Site-Specific Attribute on page 55](#).

1. In ThingWorx Composer, select **SYSTEM ► Localization Tables**.
2. For English labels, select and edit the **Default** localization table. Otherwise, select the localization table for the language you want to modify.
3. Locate the label and edit the text in the **This Language** column. For example, you could change Last Modified to Last Updated.



4. Save your changes.

## Configuring Windchill Attributes as App Properties

To add or remove Windchill attributes displayed in the app, use the following procedure.

For information on non-Windchill attributes, see [Configuring Non-Windchill Attributes as Navigate App Properties on page 59](#).

## Note

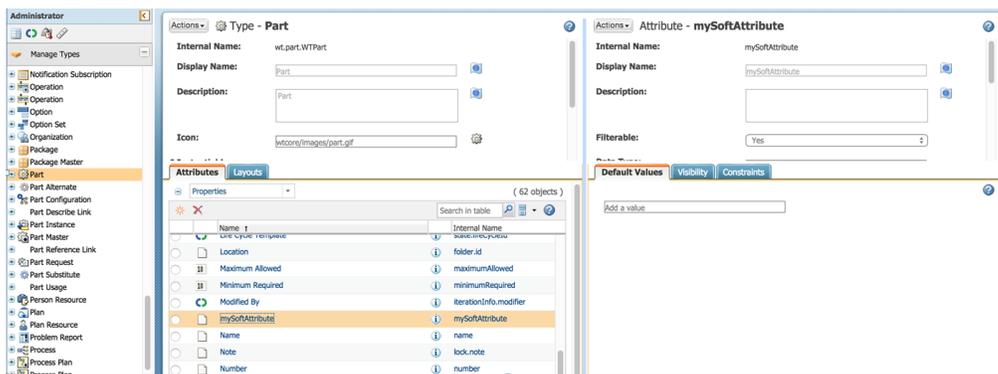
Note the following information when configuring app properties:

- You can add or remove the properties displayed by the app data shape. The standard attributes displayed at the top of app pages cannot be removed: **Number, Name, Version, and State**. You can change the label name for the standard attributes. For more information, see [Modifying a Property Label for a Default Attribute on page 51](#).
- For part apps, use attributes for part, part master, or part usage types.

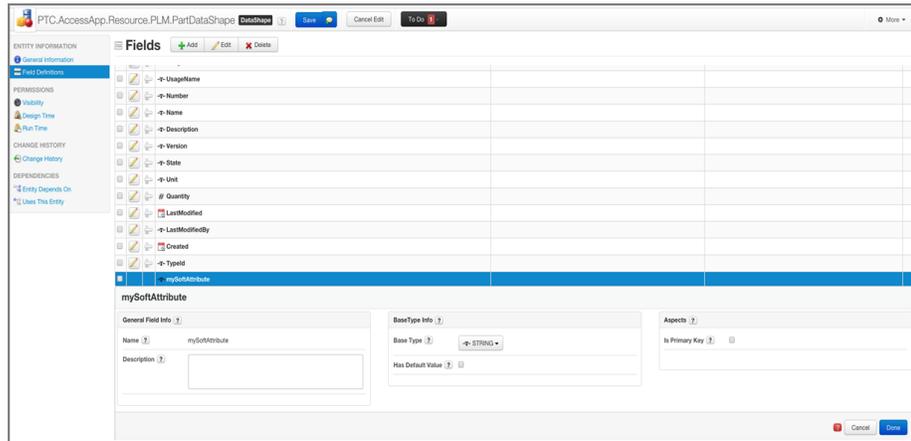
This procedure uses the **View Parts List** app as an example.

- In the Windchill Type and Attribute Management utility, identify the display name and internal name for the attribute you want to display in the app. Both will be required in the steps to follow. Here are some examples.
  - If you want to add **Modified By**, *Modified By* is the display name and `iterationInfo.modifier` is the internal name.
  - `mySoftAttribute` is an example of an attribute unique to your site. The display name and internal name happen to be the same.

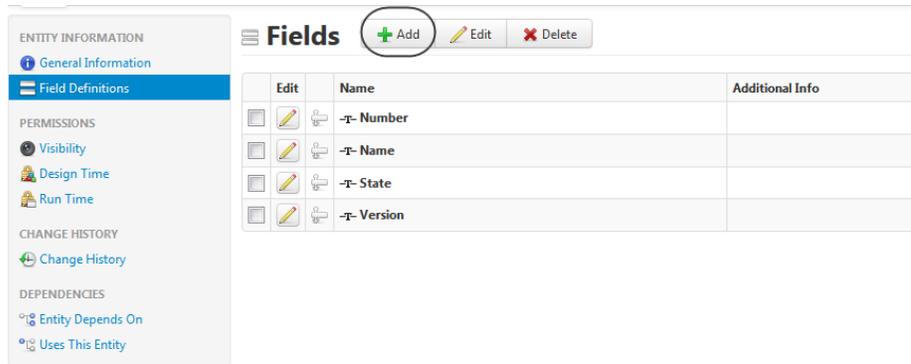
This example uses `mySoftAttribute`.



- In ThingWorx Composer, add the Windchill attribute as field definitions in the required data shapes.
  - If the attribute is unique to your site, meaning it is not included with Windchill by default, add it to **PTC.Resource.PLM.PartDataShape**.
    - Under **Entity Information**, select **Field Definitions**. Then click **Add**.
    - A **New Field** pane appears.
      - In **Name**, enter the display name for the attribute.
      - In **Base Type**, select the option associated with the attribute.
    - Click **Done**.



- b. Add the attribute to the data shape identified on the [app-specific tailoring page on page 49](#), for example, **Example1PartListAccessAppDataShape**.
  - i. Under **Entity Information**, select **Field Definitions** and then click **Add**.



- ii. A **New Field** pane appears.
  - In **Name**, enter the display name.
  - In **Base Type**, select the option associated with the attribute. For example, if you enter `Quantity`, select **Number** for the base type.
- iii. Click **Done**.

 **Tip**

To use a property label other than the internal name entered in this procedure, see [Modifying a Property Label for a Site-Specific Attribute on page 55](#).

3. In ThingWorx Composer, open the mashup associated with the app. For more information, see [App Entity Information on page 50](#).

For this example using the **View Parts List** app, open **PartsListAccessAppMashup**.

4. Select the **DynamicColumnGrid** widget, which will modify the display for the **Parts List** table in the **View Parts List** app.
  - a. Refresh the **Grid** widget using the following steps:
    - i. Delete the binding between the **DynamicColumnGrid** widget and the **GetleftnodeDecendants** service by clicking the red x icon next to **data**.
    - ii. Re-bind the **DynamicColumnGrid** widget and the **GetleftnodeDecendants** service by selecting **data** and drag-and-drop onto the **DynamicColumnGrid** widget.
  - b. Select **Configure Grid Columns**.
  - c. 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.

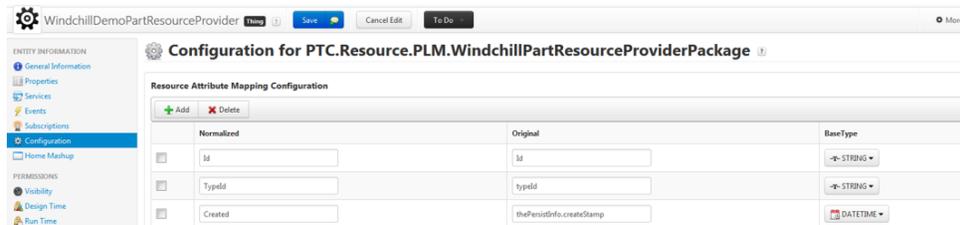
The screenshot shows the 'Configure Widget' dialog for the 'mySoftAttribute' widget. The dialog is divided into several sections:

- Attributes List:** A list of attributes on the left side, including 'mySoftAttribute', 'Quantity', 'Unit', 'LastModifiedBy', 'LastModified', 'Created', 'Description', 'HasChildren', 'Id', 'NavigationSessionId', 'ParentId', 'TreeId', 'TypeId', 'UsageId', and 'UsageName'. 'mySoftAttribute' is selected.
- Column Format:** A section with 'Column Title' set to 'PTC.AccessApp.mySoftAttribute' and 'Field Description'.
- Formatting Options:** A section with 'Auto-Width' checked, 'Width (px)' set to 120, and 'Text Alignment' set to 'Left Aligned'.
- Cell Editing Options:** A section with 'Editable' unchecked, 'Validation Expression', and 'Validation Message'.

- d. Ensure the remainder of the selected attributes match the field definitions in data shape and are configured as described in the previous step.
5. Add the attribute internal and display name to the required thing using the following steps:
    - a. For part attributes, edit **WindchillDemoPartResourceProvider** thing.  
For document attributes, edit **WindchillDocumentResourceProvider** thing.
    - b. Select **Configuration** and add the attribute.
      - **Normalized:** Enter the display name for the attribute.
      - **Original:** Enter the internal name.

If you want a localized value to display and the attribute is new, first check how the attribute value displays in the app. Some attributes automatically display the localized value.

If the localized value did not display automatically, add `.displayValue` at the end of the **Original** value.



6. Save all changes.

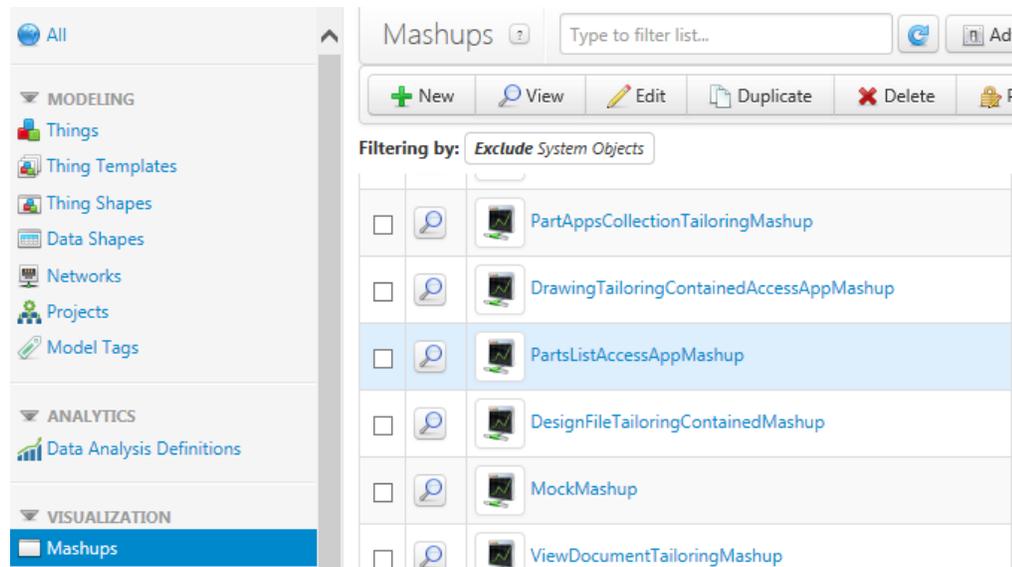
## Modifying a Property Label for a Site-Specific Attribute

This procedure describes how to modify the text of a custom attribute added to PTC Navigate by your site. To modify labels for default attributes included with PTC Navigate, see [Modifying a Property Label for a Default Attribute on page 51](#).

1. In [App Entity Information on page 50](#), note the name for the mashup and widget used by the app you want to modify.

This procedure uses the **View Parts List** app as an example.

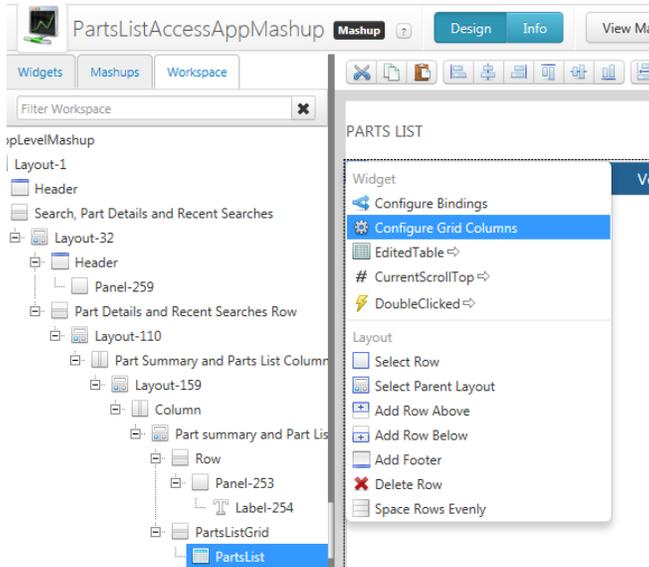
2. In ThingWorx Composer, search for the mashup, for example, **PartsListAccessAppMashup**.



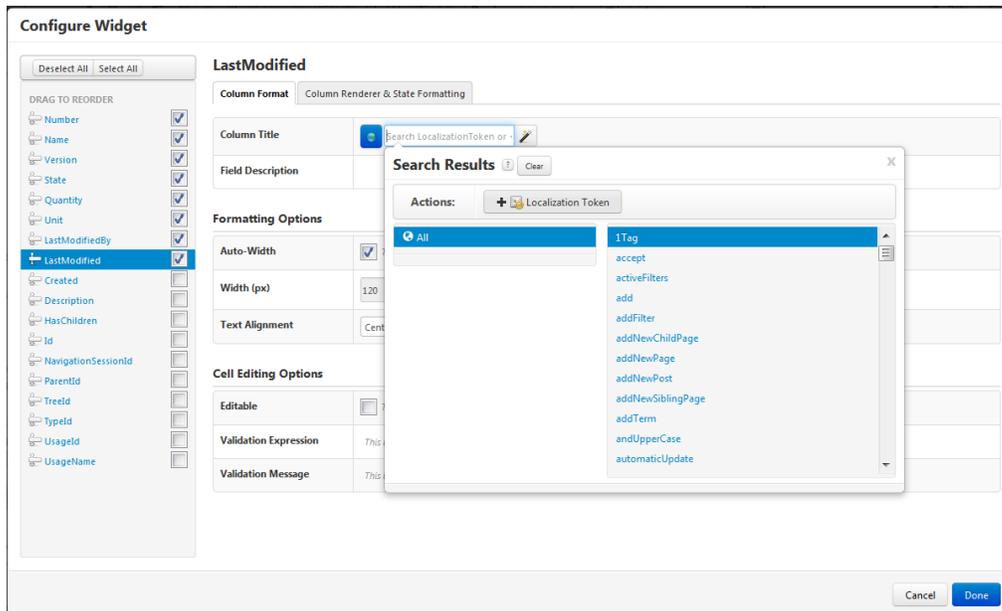
3. Edit the mashup, and click the **Workspace** tab.

a. Navigate to the widget associated with the app, for example, **PartList**.

- b. From the action drop-down for the grid widget, select **Configure Grid Columns**.



4. In the **Configure Widget** window, select the checkbox for the property label you want to modify. For this example, select **LastModified**.
5. Click the wand icon  next to the **Column Title**.



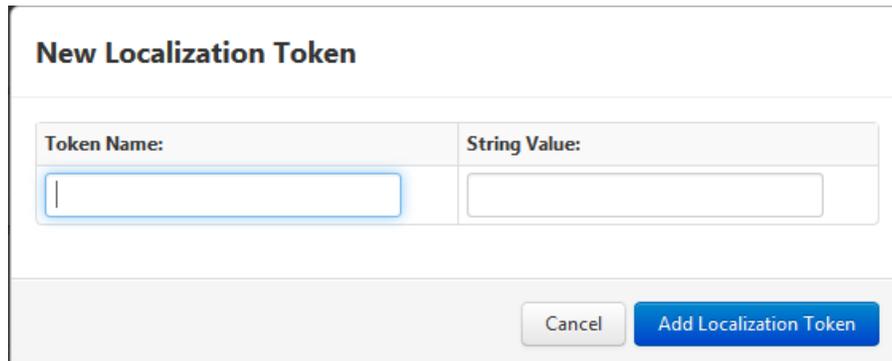
---

 **Tip**

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

---

6. Click **Localization Token**.
7. In the **New Localization Token** window, enter the following information:
  - **Token Name:** Internal placeholder for strings replaced at runtime for localization.
  - **String Value:** Label that displays in the app.



Token Name:	String Value:
<input type="text"/>	<input type="text"/>

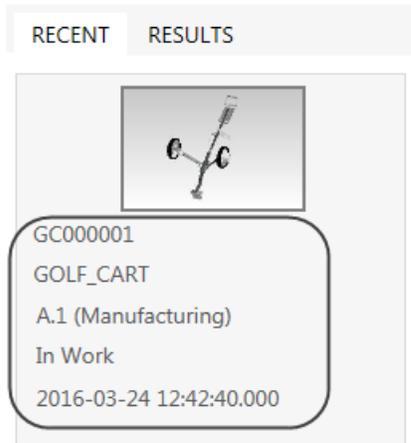
Cancel Add Localization Token

8. Click **Add Localization Token**.
9. Save the mashup.

For more information, see [Localizing Labels and Fields in Your Application](#) on page 84.

## Configuring Attributes on Recent Searches

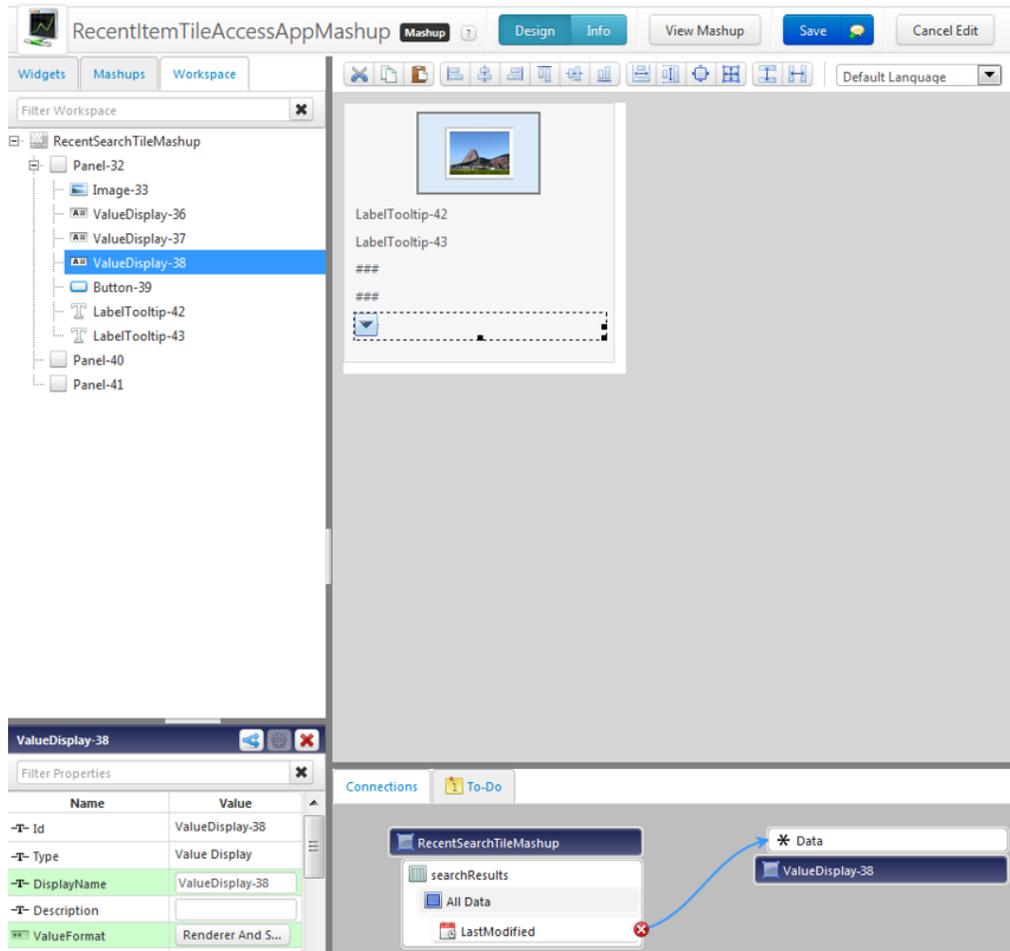
To modify the attributes that appear with items displaying under the **RECENT** tab, use the following procedure.



### Adding a New Attribute

1. Edit the recent searches data shape to add the attribute:
  - For part attributes, edit **WindchillDemoRecentSearchesPerUser**.
  - For document attributes, edit **DocumentRecentSearchesPerUser**.
2. Edit the resource data shape to add the attribute:
  - For part attributes, edit **PTC.Resource.PLM.PartDataShape**.
  - For document attributes, edit **PTC.Resource.PLM.DocumentTemplateDataShape**.
3. If the internal and display names are different, edit the resource provider thing to add the attribute:
  - a. For part attributes, edit **WindchillDemoPartResourceProvider** thing.  
For document attributes, edit **WindchillDocumentResourceProvider** thing.
  - b. Select **Configuration** and add the attribute.
    - **Normalized**: Enter the display name for the attribute.
    - **Original**: Enter the internal name.
4. Edit **RecentItemTileAccessAppMashup**.
  - a. Add **Value Display** widget on button.
  - b. From drop-down list of the newly added widget, select **Configure Bindings**.
  - c. From the **Configure Widget** window, click **Binding Sources**.
  - d. In the **Add Data Binding** window, select the new attribute you added in the previous steps.

- e. Click **Done**.
5. Verify that the attributes appear as intended. For example, in the screen below, the ValueDisplay binding with LastModified.



## Removing an Attribute

1. Edit the **RecentItemTileAccessAppMashup**.
2. Remove the binding for **Value Display** widget.
3. Save changes.

## Configuring Non-Windchill Attributes as Navigate 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.

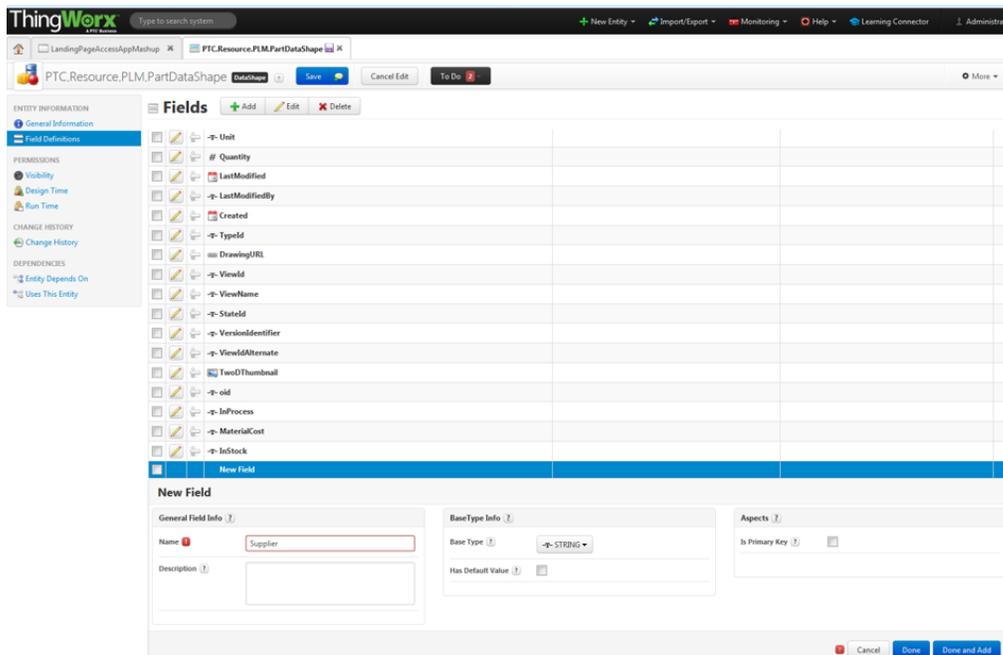
The following example uses `ERPDemoPartResourceProvider`.

Complete the steps described in the following sections:

1. Update data shapes.
2. Update related mashups.
3. Configure the composite provider.

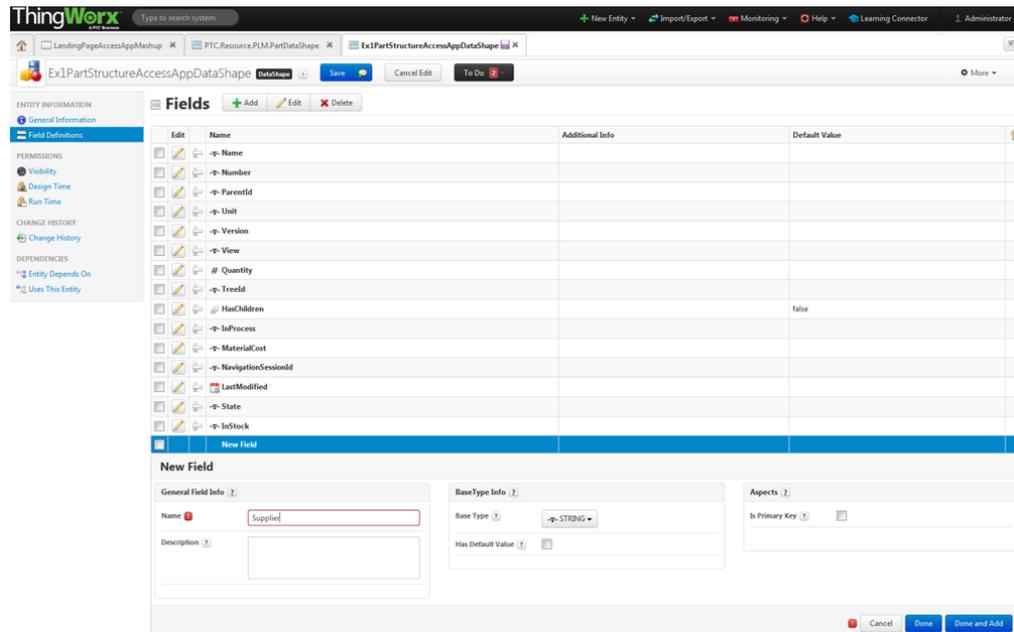
## Update Data Shapes

1. Update **PTC.Resource.PLM.PartDataShape** to add new data attributes contributed by the secondary resource providers. In this example `ERPDemoPartResourceProvider` is the resource provider and these new attributes are added: **InProcess**, **MaterialCost**, **InStock**, **Supplier**.



2. In order for the apps to display the attributes from the secondary resource provider, update the data shapes to add fields. The data shapes for each app are listed in [App Entity Information on page 50](#).

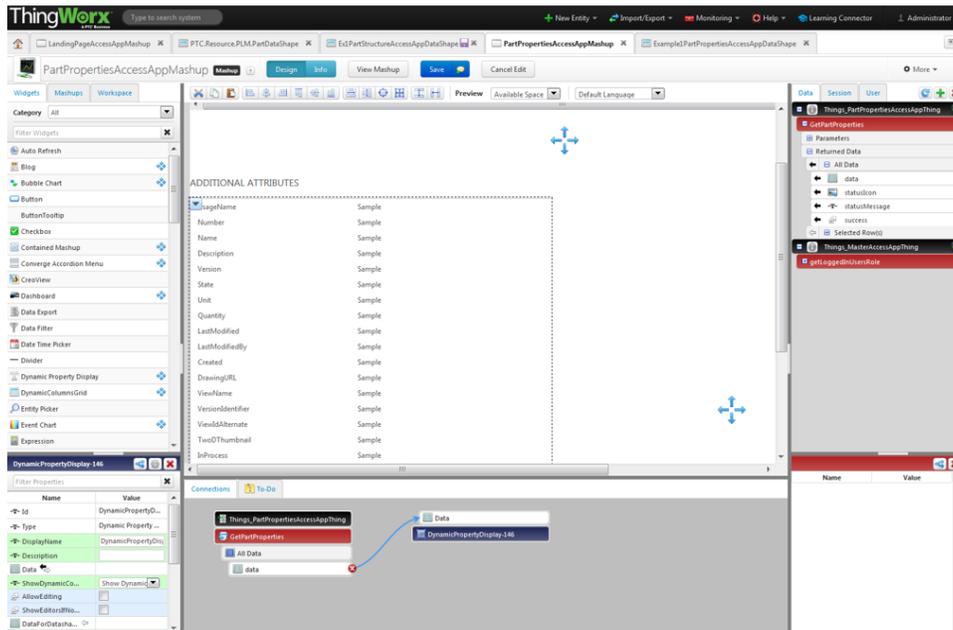
In the following example, these fields are added to **Ex1PartStructureAccessAppDataShape**: **InProcess**, **MaterialCost**, **InStock**, **Supplier**.



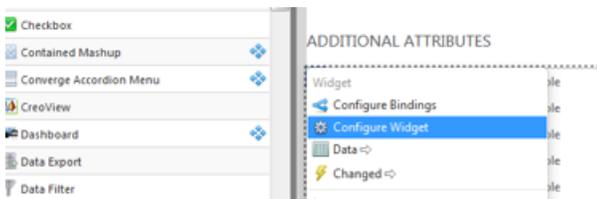
## Update Related Mashups

PTC Navigate relies on the widget configuration to make the new attributes visible in the app.

1. Edit **PartPropertiesAccessAppMashup**.
  - a. Refresh the **ADDITIONAL ATTRIBUTES** widget using the following steps:
    - i. Delete the binding between the **DynamicPropertyDisplay** widget and the **GetPartProperties** service by clicking the red x icon next to **data**.
    - ii. Re-bind the **DynamicPropertyDisplay** widget and the **GetPartProperties** service by selecting **data** and drag-and-drop onto the **DynamicPropertyDisplay** widget.



- b. From the **ADDITIONAL ATTRIBUTES** drop-down arrow, select **Configure Widget**.



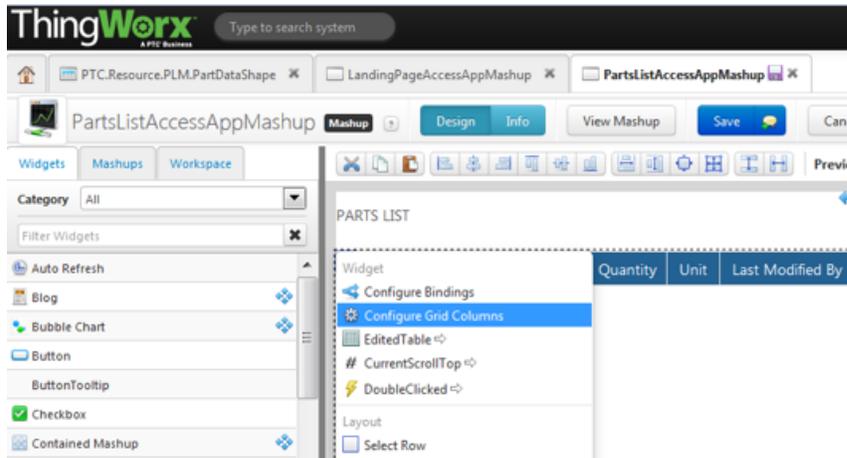
- c. In the **Configure Widget** window, select the attributes from `ERPDemoPartResourceProvider` that you want visible in the app.
- d. Click **Done** and save the mashup.
- e. A known issue for this release is that after re-binding the **DynamicPropertyDisplay** widget and the **GetPartProperties** service, the localization tokens on each widget column are lost. Assign each localization token in the **Configure Widget** window using the information in the following table. For more information, see [Localizing Labels and Fields in Your Application on page 84](#).

Column Name	Localization Token for Column Title
Last Modified By	PTC.AccessApp.LastModifiedBy
Description	PTC.AccessApp.Description
Unit	PTC.AccessApp.Unit
Name	PTC.AccessApp.Name
Created	PTC.AccessApp.Created

Column Name	Localization Token for Column Title
Last Modified	PTC.AccessApp.LastModified
Number	PTC.AccessApp.Number
Version	PTC.AccessApp.Version
Quantity	PTC.AccessApp.Quantity
View Name	PTC.AccessApp.ViewName
State	PTC.AccessApp.State
Usage Name	PTC.AccessApp.UsageName

2. Edit **PartsListAccessAppMashup**.

- a. Select the **PartsList** widget and from the drop-down arrow, select **Configure Grid Columns**.



- b. In the **Configure Widget** window, select the attributes from `ERPDemoPartResourceProvider` that you want visible in the app.
- c. Click **Done** and save the mashup.

3. Edit **PartStructureAccessAppMashup**.

- a. Select the **PART STRUCTURE** widget, and from the drop-down arrow, select **Configure Tree Table Columns**.
- b. In the **Configure Widget** window, select the attributes from `ERPDemoPartResourceProvider` that you want visible in the app.
- c. Click **Done** and save the mashup.

## Configure the Composite Provider

The **Use Composite Provider** option is hidden from the app tailoring pages by default. An administrator must have this option available for the attributes from the `ERPDemoPartResourceProvider` to appear in the app.

1. In each of the following mashups, select the **Use Composite Provider** label and in the lower left corner of the screen, select the **Visible** checkbox.

The screenshot displays the ThingWorx user interface for configuring a mashup. The main window is titled "PartPropertiesTailoringContainedMashup" and shows a "Design" view. On the left, a "Widgets" panel lists various components, with "Checkbox" selected. Below this, the "Properties" panel for the "Checkbox-123" widget is visible, and the "Visible" property is checked. The central workspace shows the configuration for the "Use Composite Provider" option, which is set to "Yes". Below this, there are several input fields for validation and configuration, including "CallIsDirtyValidatorOut", "CallGetConfig Validator Out", "isPageDirty", and "notificationVisibleFromSetConfig". At the bottom, a "Connections" panel shows a flow diagram with a "GetConfiguration" action connected to a "Checkbox-123" widget, and a "useCompositeProvider" action connected to the "State" property of the widget.

- 
- **PartPropertiesTailoringContainedMashup**
  - **PartsListTailoringContainedMashup**
  - **PartStructureTailoringContainedMashup**
2. Go to the tailoring page for any one of the applicable apps, and select the **Use Composite Provider** checkbox.
- **View Part Properties**
  - **View Parts List**
  - **View Part Structure**



# 4

## PTC Navigate View ALM App Extension

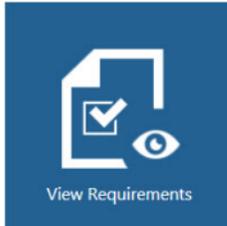
PTC Navigate View ALM App Extension Product Overview .....	68
Product Requirements for PTC Navigate View ALM App Extension .....	68
Importing the PTC Navigate View ALM App Extension .....	69
Configuring PTC Navigate View ALM App Extension.....	70
Tailoring PTC Navigate View Requirements App.....	73
Known Limitations in PTC Navigate View Requirements App.....	75

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

---

## 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.



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 Requirements App on page 73](#).

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.

---

## Product Requirements for PTC Navigate View ALM App Extension

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

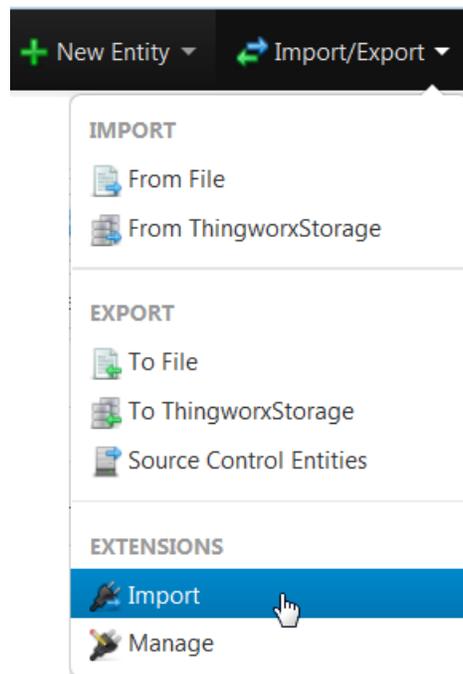
- Integrity at 10.8 or later
- For other platform support, see [Product Requirements for PTC Navigate 1.2 on page 8](#).

---

# Importing the PTC Navigate View ALM App Extension

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_ILM_Connector_ExtensionPackage.zip`.

---

### **Note**

This release of PTC Navigate View Requirements 1.1 is compatible only with ILM Connector Extension 2.0, which is a part of the installation package of this release.

Before you import this extension, ensure that you uninstall previous versions of PTC Navigate View ALM App Extension and ILM Connector Extension.

If the PTC Integrity Extension for ThingWorx needs to be imported, it is recommended to be done on a separate ThingWorx installation different from the one used for PTC Navigate to avoid possible compatibility issues.

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**.

## Configuring PTC Navigate View ALM App Extension

This chapter contains configuration information for ThingWorx and PTC Integrity Lifecycle Manager. The configuration steps explained here are also useful when single sign-on authentication is used.

### 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 Integrity. For more information, see [Configuring Single Sign-On when Connecting to Windchill on page 35](#).

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 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 Integrity is done by impersonation where administrator permissions are required for impersonation. In this case, the Integrity Connector retrieves documents from 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 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 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**.

## Modifying 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:

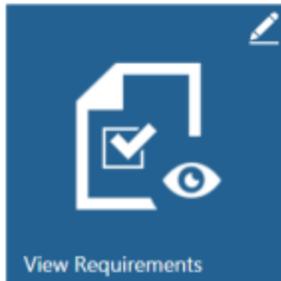
1. In the ThingWorx Composer, under **Things**, click the **PTCRODocViewAccessAppThing** thing. The mashup for the thing opens.
2. Under **Permissions**, select **Run Time** type of permission.
3. Grant Allow permissions to **Property Read** and **Service Execute**.
4. In **Group or User**, add the required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

## Tailoring PTC Navigate View Requirements App

### 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

**CONFIGURATIONS**

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

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 configuration 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.

---

 **Note**

If you are upgrading from PTC Navigate View ALM App Extension 1.0 to PTC Navigate View ALM App Extension 1.1, ensure that the Administrator takes note of the existing configuration table from the existing server and manually updates it on the new server.

---

### Removing a Configuration

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

## Known Limitations in PTC Navigate View Requirements App

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.

---

### **Non-numeric Document ID not Supported**

You cannot type a non-numeric document ID the Search field while trying to access documents from PTC Integrity Lifecycle Manager.



# 5

## PTC Navigate Reference Information

Advanced Customization .....	80
Example Configuration Using SSL for Secure Communications .....	96
Architecture of Single Sign-On.....	99

This section provides advanced customization information, a configuration example, and an overview of the single sign-on architecture.

---

## Advanced Customization

This section 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

## Adding Apps to the Navigate Landing Page

The following sections describe important components required when adding apps to the landing page. Sample steps are included toward the end of this topic.

### Data Shapes

The framework defines two data shapes:

- **PTC.NavigateApps.AppDataShape:** Includes the attributes required to identify an app as a Navigate app.
  - **thingName:** The name of the thing that signifies a Navigate app. The permissions set on this thing will control access to the app for a given user. If the access permissions don't exist for a given user, an app will not be displayed on the landing page for this user.
  - **mashup:** The name of the mashup for the app.
  - **url:** The relative url to launch the app. For example:  
`index.html#mashup=CommonSearchAccessAppMashup&configEntity=Example1AccessAppConfigThing&...`
  - **tailoringMashup:** The name of the tailoring mashup for the app, if one exists.
  - **tailoringUrl:** The relative url, if one exists. For example:  
`index.html#mashup=DrawingTailoringAccessAppMashup&accessAppConfigThingName=Example1DrawingAccessAppConfigThing&...`
  - **displayName:** Display name for the app.
  - **appIcon:** The icon for the app displayed on the landing page.
- **PTC.NavigateApps.CollectionDataShape:** Includes the attributes required to identify a collection of Navigate apps.

- An infotable of apps where each row is an app that conforms to **PTC.NavigateApps.AppDataShape**.
- A name for the collection: The name of the tailoring mashup, if one exists.
- The relative url if one is relevant for the collection. For example: index.html#mashup=PartAppsCollectionTailoringMashup.
- Display rank: Collections display on the landing page in ascending order of the rank.

## Thing Shape

**PTC.NavigateApps.AppCollectionProviderThingShape**: This thing shape exposes a service called **GetAppCollections**. All app collection providers should declare that they implement this thing shape. The framework's collection consolidator depends on this declaration to identify all providers from which to collect Navigate app collections for display on the landing page. In addition, **DefaultAppCollectionProvider** is available. This class implements a service handler for the thing shape's service.

## Classes, Interfaces

- **INavigateApp**: An interface that should be implemented by a Navigate app thing. This thing provides a definition for what constitutes a Navigate app. This interface is complimentary to the **PTC.NavigateApps.AppDataShape** definition and forces the things that implement it to provide the attributes required by that data shape. In addition, the permissions set on the things implementing this interface will control access to the app for each user. If the access permissions don't exist for a user, an app will not display on the landing page for this user.
- **AbstractAppCollectionProvider**: An abstract class that should be extended to things in an extension (app collection providers) that are responsible for contributing a collection of **INavigateApps**. This class defines a **GetAppCollections** service that each app collection provider should implement. The purpose of this service is to define the logic for how an extension would collect a set of Navigate apps that it will contribute to the framework. In addition to implementing this service, the app collection providers should declare that they implement **PTC.NavigateApps.AppCollectionProviderThingShape** to help the framework's app collection consolidator lookup the providers that provide this service.
- **AppCollectionsConsolidator**: The main responsibility of this class is to look up the app collection providers and call on those providers to consolidate app collections. In addition, this class provides the **addAppCollectionRow** utility service. The providers can use this service to pool a set of apps given a criteria (a **TagLink**) to look up apps. For example, if **INavigateApp** apps are tagged as

---

PTC.AccessApp.PartApps, then `addAppCollectionRow` can be invoked by providing, among other attributes, `PTC.AccessApp.PartApps TagLink` as the criteria that should be used to collect the apps as well as an instance of the `InfoTable` in which to add a new collection row.

The following example shows the minimal steps an extension performs to display apps on the landing page. These steps enable an extension to contribute app collections to the framework.

1. Add dependency on `TWX_Converge_Core_ExtensionPackage` and `PTCNavigate_FrameworkExtensionPackage` artifacts in your extension package. See the following example:

```
<dependency>
  <groupId>com.ptc.cpm.core</groupId>
  <artifactId>TWX_Converge_Core_ExtensionPackage</artifactId>
  <version>7.0.1-1133</version>
</dependency>

<dependency>
  <groupId>com.ptc.converge.apps.access</groupId>
  <artifactId>PTCNavigate_FrameworkExtensionPackage</artifactId>
  <version>1.0.0-SNAPSHOT</version>
</dependency>
```

2. Create `Navigate` app things that implement `INavigateApp`. Tag these things appropriately to identify in which collection the apps belong. See the following example:

```
public class AccessAppConfig extends Thing implements INavigateApp
```

3. Create a `Navigate` app collection provider thing that extends `AbstractAppCollectionProvider` and provides an implementation for `GetAppCollections`. Use the `addAppCollectionRow` utility method in `AppCollectionsConsolidator` to pool apps into collections. See the following example:

```
public class MasterAccessApp extends AbstractAppCollectionProvider
```

```
/* (non-Javadoc)
 * @see com.ptc.converge.apps.access.AbstractAppCollectionProvider#GetAppCollections()
 */
@Override
public InfoTable GetAppCollections() throws Exception {
    InfoTable consolidatedAppCollections = InfoTableInstanceFactory
        .createInfoTableFromDataShape("PTC.NavigateApps.CollectionDataShape");

    // collect part apps
    appCollectionConsolidator.addAppCollectionRow(consolidatedAppCollections, "Part Apps"
        "index.html#mashup=PartAppsCollectionTailoringMashup", "PartAppsCollectionTailoringMashup",
        new TagLink(VOCAB_PTC, PART_APPS_TAG), 1);

    // collect doc apps
```

```

    appCollectionConsolidator.addAppCollectionRow(consolidatedAppCollections,
        new TagLink(VOCAB_PTC, DOCUMENT_APPS_TAG), 2);
    return consolidatedAppCollections;
}

```

4. Declare that the Navigate app collection provider thing implements **PTC.NavigateApps.AppCollectionProviderThingShape**. See the following sample ThingWorx thing configuration:

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Entities build="b456" majorVersion="6" minorVersion="0"
    revision="2" schemaVersion="803">
    <Things>
        <Thing aspect.isEditableExtensionObject="true" description="" document
            effectiveThingPackage="PTC.Converge.App.MasterAccessAppPackage"
            enabled="true" homeMashup="" identifier=""
            lastModifiedDate="2015-08-05T23:40:58.139-04:00" name="MasterAc
            published="false" tags="PTC:PLM;PTC:PTC.AccessApp.Master"
            thingTemplate="PTC.Converge.App.MasterAccessAppTemplate" valueS
            <ImplementedShapes>
                <ImplementedShape name="PTC.NavigateApps.AppCollectionProvid
                <ImplementedShape name="PTC.Capability.ResourceServiceProvid
            </ImplementedShapes>
        </Thing>
    </Things>
</Entities>

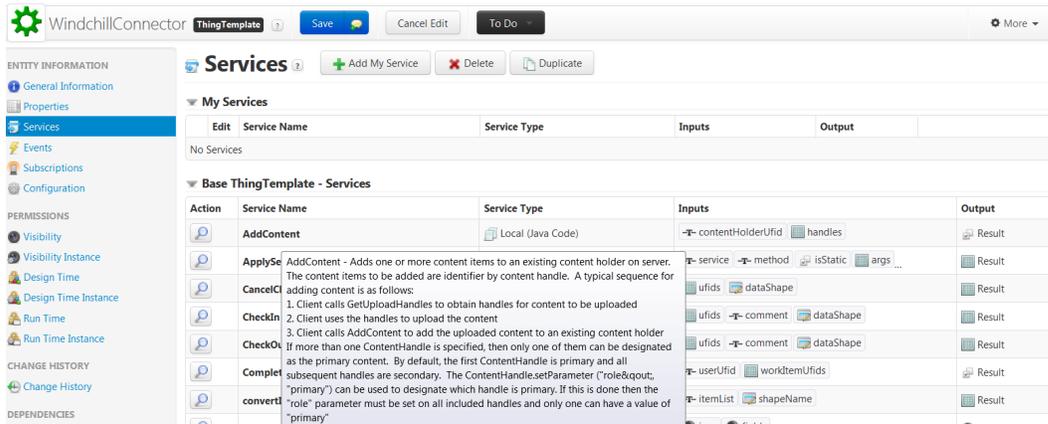
```

5. Bundle `PTCNavigate_FrameworkExtensionPackage.zip` along with your app extension package ZIP so both package ZIP files get imported together into ThingWorx.
6. Import the bundle and launch `LandingPageAccessAppMashup` to see your app on the landing page.

## Entity Overview

The 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.

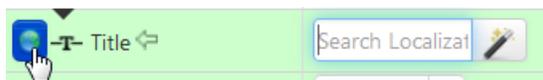
After installation and configuration are complete, learn more about the **WindchillConnector** thing template by opening ThingWorx Composer and viewing its list of services. As shown in the screen below, you can hover over a service to view additional information about the service.



## 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 Windchill Extension entities.

To localize elements in entities, you must include a token for each element. Click the localization toggle icon to search for and select an existing token or create a new token for a field.



All tokens that you create are added to the localization tables maintained by ThingWorx. By creating and populating localization tables, you can display runtime elements in different languages. For details on how to create and populate localization tables, search for “localization tables” in the ThingWorx help center. The help center is available from the [PTC Help Centers page](#).

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

<http://youtu.be/3bo6HtXSqRA>

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

## Adding Saved Filters to Your Mashup

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

---

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

- In a part structure mashup, a filter can only be specified by name. Ensure that all filters that you want to use in a ThingWorx application have unique names.
- In most instances, saved filters are intended to be used with the specific object type for which they were created. Ensure the selected filters work with the object structure you want to display in the ThingWorx application.

## PTC Navigate

**Saved Filter** options are available on the app tailoring pages. For more information, see the “Common Tailoring for Apps” section.

## Windchill Extension

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



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

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

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

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

## Using Creo View Extension

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

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

- 
- Change the transparency and color of the Creo View parts.
  - Change the background color of the Creo View plugin.

## Properties and Events

Creo View Extension has the following properties and events:

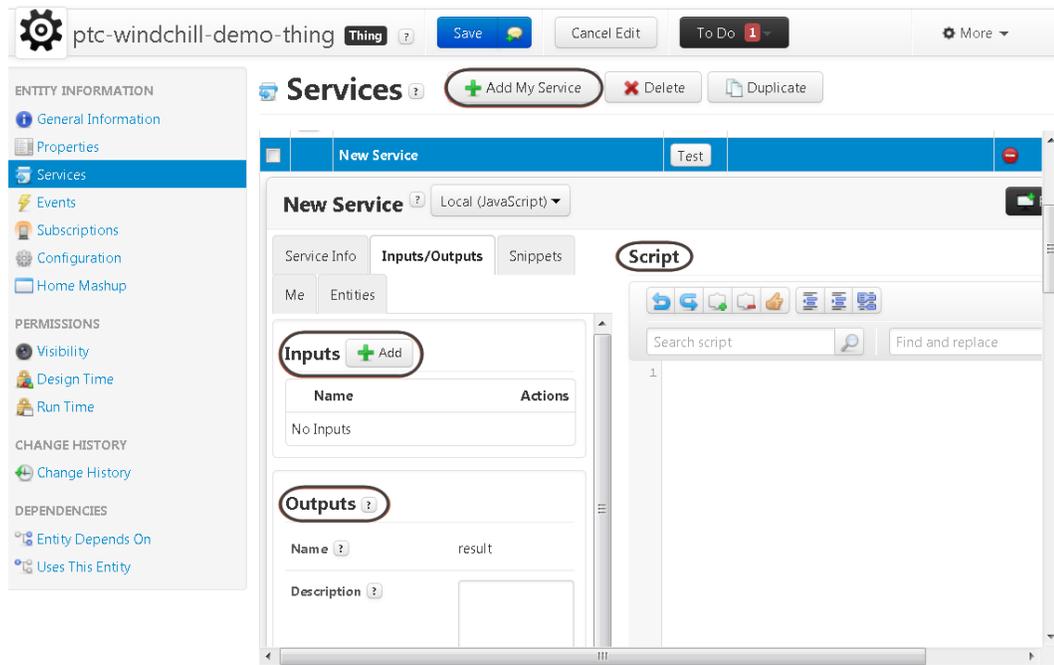
- `ProductToView` – This property enables you to visualize 3D data from Windchill in the Creo View Extension. It is the URL to the Creo View data file. Populate the `ProductToView` property to visualize the data. Windchill sample mashups provided with the installation demonstrate how to use sample services to populate this property.
- `BackgroundStyle` – This property enables you to change the background color of the Creo View 3D graphics area.
- `Data` – This property enables you to select 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 Creo View 3D data.
- `DataFormatter` – This property enables you to specify rules that must be used to apply color and transparency to the Creo View parts. Specify a column for these rules from the data table which is bound to the `Data` property.
- `OccurrenceField` – This property enables you to select the column that must be used as the Occurrence ID. The column is selected from the data table which is bound to the `Data` property.
- `selectedOccurrencePath` – This property specifies the Occurrence ID of the part selected in the 3D data.
- `selectionChanged` – This event indicates a change in the selection of 3D data.

## Using Info\*Engine Tasks when Developing Your Application

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

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

To use the **ExecuteTask** service, create a thing using the **WindchillConnector** thing template. For example, you can add a new service in **ptc-windchill-demo-thing** as shown in the following screen.



The following sections describe some common application features that you can include using Info\*Engine tasks: [Querying WTParts By Name on page 87](#) and [Using Custom Reports from Windchill on page 92](#).

## Querying WTParts By Name

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

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

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

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

<ie:webobject name="Query-Objects" type="OBJ">
  <ie:param name="INSTANCE" data="$(@FORM[]supporting-adapter[*])"
    delim="!" valueSeparator="!"
    default="<%=com.infoengine.au.NamingService.getVMName() %>" />
</ie:webobject>
```

---

```
<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 Parameter:**

STRING name

### Input Parameter

Name ?	<input type="text" value="name"/>
Description ?	<input type="text"/>
Base Type ?	<input type="text" 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 could contain line numbers, hidden editing characters (such as tabs and end-of-line characters), and extraneous spaces. If you cut and paste code, check for these characters and remove them before attempting to use the example in your application.

---

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

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

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

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

3. Save the service.

---

## Using Custom Reports from Windchill

---

### Note

For more information on custom reports, see the Windchill Help Center or *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 does:

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

### Before You Begin

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

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

---

### Note

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

---

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

---

## Implementing the Service

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

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

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

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

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

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

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

```

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

```

## Modifying Windchill to Use the Service

Before you can use your service, you must modify Windchill so that it allows HTTP access to the `ExecuteReportTemplate` task. To do this, add `access=http` to the `.delegateInfo` files. Add this file to the task folder, `tasks\com\ptc\windchill\enterprise\report`, or any parent folder.

## 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 Windchill Extension includes the **WindchillConnector** thing template that provides the **processJSONRequest** service used to invoke Windchill REST web services.

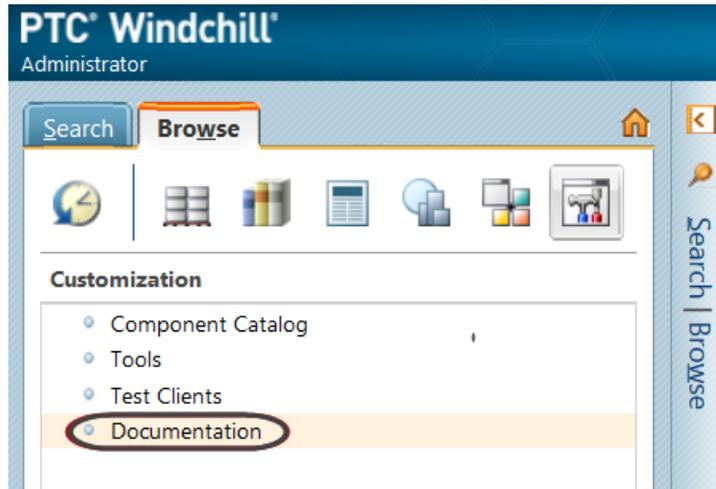
To invoke a Windchill REST service, you can use the **processJSONRequest** service that is available from **WindchillConnector** thing template. For example, the following code snippet from **ptc-windchill-demo-thing** invokes the Windchill service `/structure/objects`, passing parameters that specify 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		
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 Windchill Help Center.

## ServiceTimeout Property

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

```
var params = {
```

```

queryParams: {
    '$select': 'name,number,version'
},
data: {
    keyword: undefined
},
type: 'POST',
url: me.getRestUrl() + "/search/saved-searches/" + encodeURIComponent(oid) + "/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 'PTC Connection Settings' table is as follows:

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

## Example Configuration Using SSL for Secure Communications

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

---

 **Note**

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

1. Create a truststore for ThingWorx and import the Windchill certificate:
2. Create keystore for ThingWorx.
3. Configure Windchill to trust ThingWorx.

---

 **Note**

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

- a. Update PTC HTTP Server configuration to reference the CA certificates file.
  - i. Navigate to and edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.
  - ii. Uncomment the `SSLCACertificateFile` entry so that it refers to the `ca-bundle.crt` file. Note the location of the `ca-bundle.crt` file.
  - iii. Save changes.
- b. Add the ThingWorx certificate to PTC HTTP Server's list of trusted CA certificates.

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

---

 **Note**

If the file does not yet exist, create the file and any required directories.

---

- ii. Append the contents of the ThingWorx certificate (`servercert.pem` from step 2a above) held in the trust store to this file.
  - iii. Save changes.
- c. Configure PTC HTTP Server to export the certificate information to Windchill.
    - i. Edit `<HTTPSERVER_HOME>/conf/extra/httpd-ssl.conf`.
    - ii. Find the line with `SSLOptions`.
    - iii. Uncomment out the line with `SSLOptions +ExportCertData`.

For example:

```
SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>
<Directory "C:/ptc/Windchill_10.2/HTTPServer/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>
```

- iv. Save changes.
- d. Configure PTC HTTP Server for the `sslClientAuth` URL.
    - i. Find the Windchill web app name in `<windchill-home>/codebase/wt.properties` in the entry `wt.webapp.name`.
    - ii. Open a shell or command prompt and enter the following:

```
cd <HTTPSERVER_HOME>ant -f webAppConfig.xml -DappName=[windchill-web-app]
-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</filtername>`.
      - 3) Add another `<init-param>` element after the existing one where `[thingworx-cert-name]` is the name of the ThingWorx certificate.

---

For a self-signed certificate, the certificate name is the Owner value for the certificate when running `keytool -list -v -keystore <path-to-ThingWorxKeyStore.jks>`:

```
<init-param><param-name>trustedSubjectPattern.1
    </param-name><param-value>[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 that no configuration errors exist.
5. Launch **ptc-connected-demo-mashup** to verify the ThingWorx – Windchill trusted SSL configuration.

## Architecture of Single Sign-On

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

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

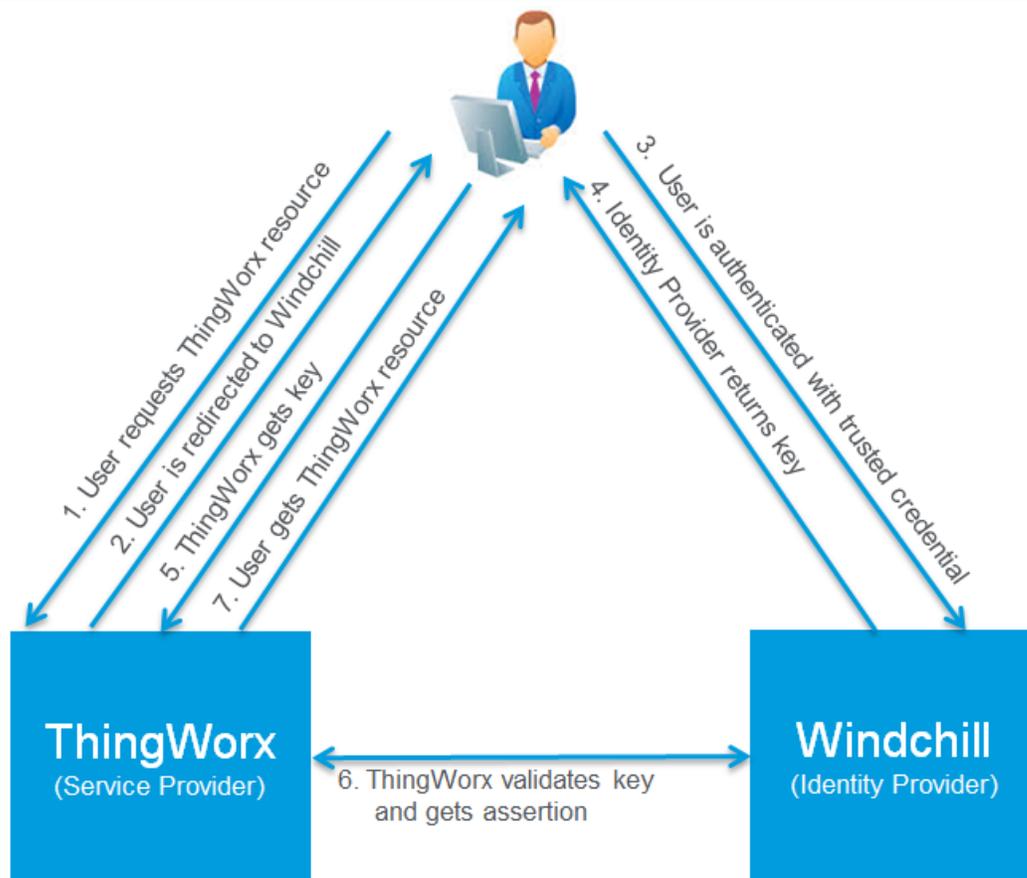
The architecture consists of two components:

- PTC Identity Provider Authentication Filter – Tomcat filter for authentication; the filter redirects unauthenticated users to the IdP and validates the key.
- PTC Identity Provider Authenticator Extension – ThingWorx extension for authentication; the extension includes dynamic user creation and setting home mashup.

## Design of Authentication Filter

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

The following diagram shows the authentication flow:



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

1. User attempts to access the ThingWorx application.
2. The PTC Identity Provider Authentication Filter intercepts the ThingWorx user request and redirects it to IdP for login.  
User sees Windchill login form instead of ThingWorx login form.
3. User enters user name and password in the login form. Windchill authenticates the user.
4. After a successful login, Windchill server redirects request back to ThingWorx with the generated key.

- 
5. The PTC Identity Provider key validation filter reads the key and validates it.
  6. The validation filter passes the Windchill user name to the next filter as request attribute. In the single sign-on process, the next filter is the ThingWorx Authentication Filter.

Authentication is complete.

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

## 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 36](#).

- 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 36](#).