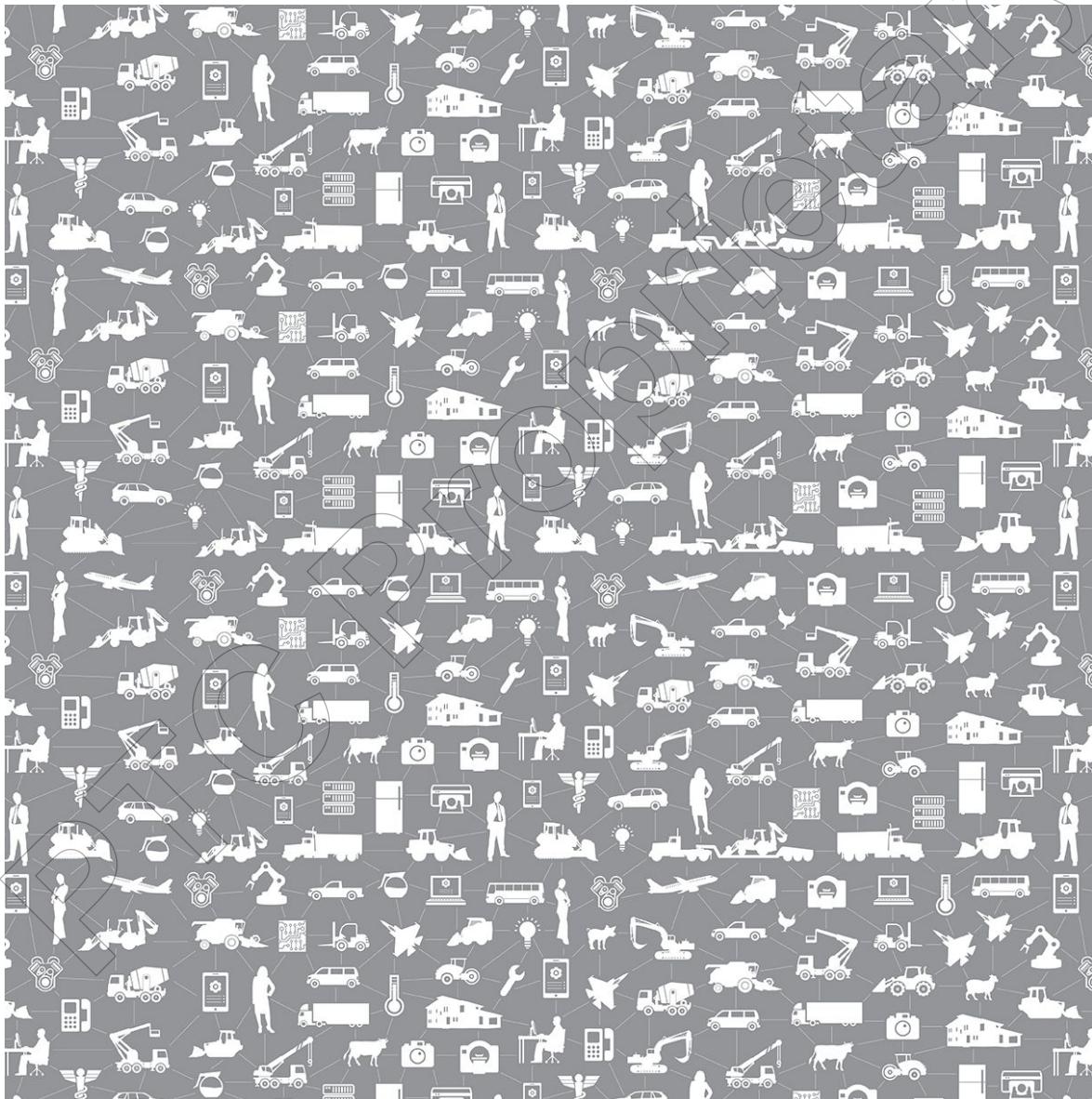


ThingWorx Manufacturing App – Custom App Development



Copyright © 2019 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

PRINTING HISTORY**Document No.**

TE-00040

Printed in the U.S.A.

Date**Description**

Initial Printing of:

ThingWorx Manufacturing App – Custom App Development

PTC Proprietary

Table of Contents

ThingWorx Manufacturing App – Custom App Development

Introduction	1-1
Creating and Accessing Cloud VM Instance - for PTC internal training only	1-2
Manufacturing Apps Overview.....	2-1
Exploring the ThingWorx Manufacturing Apps Features	2-2
Manufacturing Apps - Connectivity	4-1
Modeling Plant Assets in ThingWorx Industrial Connectivity	4-2
Configure KEPServerEX connectivity with ThingWorx Manufacturing Apps	4-4
Create the equipment structure in ThingWorx Manufacturing App	4-6
Configure the Status Configuration Expressions and Performance Metrics for each Asset.....	4-8
Customizing ThingWorx Manufacturing Apps	6-1
Customizing the Quality Calculations	6-2
Visualizing the Customized Quality Calculations in ThingWorx Manufacturing Apps	6-5
Customizing Controls Advisor.....	6-6
Other Customization Entities.....	7-1
Writing Custom Data Shapes and Services	7-2
Creating a New Custom Manufacturing App	7-5
Creating Custom Roles.....	7-12

PTC Proprietary

Module 1

Introduction

Module Overview

Welcome to the class! This module is a discussion about course logistics and expectations. Please review them with your instructor.

Objectives

After completing this module, you will be able to:

- Describe classroom logistics.
- Provide an overview of Tools and System Logistics.

Exercise 1: Creating and Accessing Cloud VM Instance - for PTC internal training only

Objectives

After successfully completing this exercise, you will be able to:

- Create VM instance for PTC internal training

Task 1: Create VM instance for PTC internal training.



If you are a PTC partner and attending training from the PTC, contact your Instructor for the VM instance.



Note that images are for reference only.

1. Browse [Technical Enablement Instructor Zone Page](#) and logon using your credentials.
2. Search your Course ID and note the VM template name mentioned in front of 'PTC Internal – Cloud Portal' under the VM Information column.

Example:

for course ID TE-00040, 'ThingWorx Manufacturing Apps 8.3 – Training VM-1 – Windows' is the VM template name.

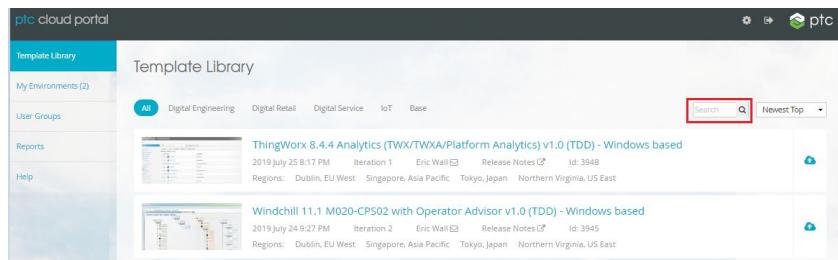
ILT/Virtual ILT Training Delivery Instructions

Select the details link for the course (ID) you are scheduled to deliver to find all necessary information on how to prepare, deliver and assess the course.

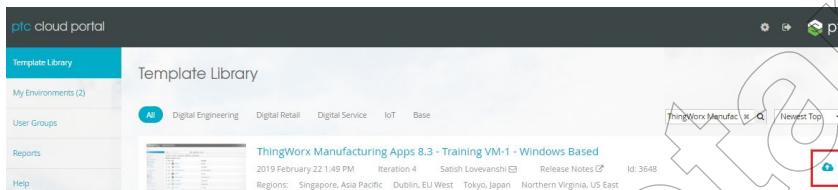
#	Course/Assessment	Course ID	Format	Duration (Hours)	Details	Curriculum/s	VM Information
4	ThingWorx Manufacturing Apps – Custom App Development	TE-00040	Virtual ILT	8	Link	ThingWorx SCO C1 Curriculum for	<ul style="list-style-type: none"> PTC Internal – Cloud Portal: ThingWorx Manufacturing Apps 8.3 – Training VM-1 – Windows PTC Partners – Azure: ThingWorx_Manufacturing_Apps_8.3_SCO-02

3. Browse to the **PTC Cloud portal** using <https://portal.ptc.io> and sign in using your credentials.

4. Go through the usage policies if you are signing in for the first time and click **Agree**.
5. Ensure that the Template Library is selected in the left panel. It lists the available templates in the portal.
6. Type/paste the template name in search field as shown in below image and click search.



7. As the ThingWorx Manufacturing Apps 8.3 – Training VM-1 – Windows template is displayed, click the **Create Environment** icon as shown in the below image.



8. Create a New Environment window as displayed in the figure. Then provide the details that are highlighted in the below image. For example:

- A. Name – For example, SCO Training
- B. Business Rationale – Training
- C. Region – Singapore, Asia Pacific (select the Region as per your geographic location)
- D. Time Zone
- E. Daily Shutdown Time
- F. Set **Daily Startup and Shutdown** time. Select the **Enable** checkbox to automatically start the VMs at the given time. Startup time should be 30 min before the scheduled class/training time.

Finally, click the **Create New Environment** button.

Create New Environment

Name: (highlighted by arrow)

Size: Medium (2CPU, 8GB RAM) (highlighted by arrow) Large (4CPU, 16GB RAM)

Business Rationale

Purpose: (highlighted by arrow) Notes:

Region: (highlighted by arrow) Promote better performance by choosing a location close to your users

Start / Stop Settings

Timezone: (highlighted by arrow)

Options

Schedule: Weekdays Everyday 24 / 5 24 / 7
Monday–Friday, scheduled daily shutdown (optional scheduled daily startup)

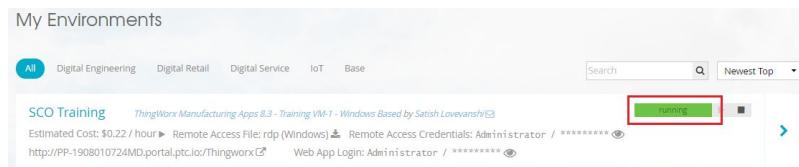
Daily Shutdown: (highlighted by arrow) PM AM Enable

Daily Startup: (highlighted by arrow) AM Enable

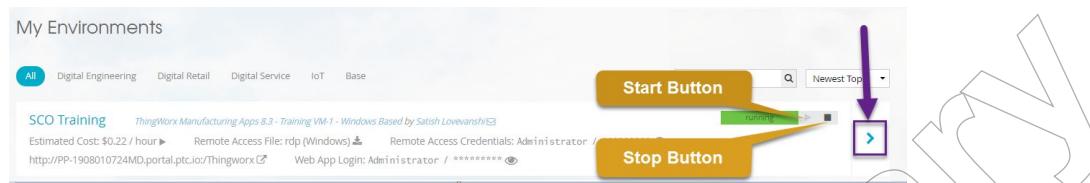
Estimated Cost:

Create New Environment (highlighted by arrow) Cancel

9. A new environment is generated and is listed under My Environments. It will take approximately 15 mins to start. Once VM starts, its turns green with status running.



You can start and stop the VM environment using the buttons as shown in the below image.



- Click on the arrow '>>' icon as shown by arrow in the image above and refer to the **Environment Access** section in the image below.

Environment Access

Remote Access File	rdp (Windows)
Remote Access Credentials	Administrator / *****
Web Application URL	http://PP-1908010724MD.portal.ptc.io/Thingworx
Web Application Credentials	Administrator / *****

The following are the details of the Environment Access:

A	Remote Access File	Download the rdp file
B	Remote Access Credentials	rdp credentials
C	Web Application URL	Web Application URL such as ThingWorx or Windchill
D	Web Application Credentials	Web Application Credentials

- Download the RDP file and open it in a text editor such as Notepad ++. Edit the value of 'prompt for credentials' setting from 0 to 1. Save the file.

```

20 keyboardhook:i:2
21 negotiate security layer:i:1
22 prompt for credentials:i:1 
23 redirectclipboard:i:1
24 redirectcomports:i:0
25 redirectposdevices:i:0

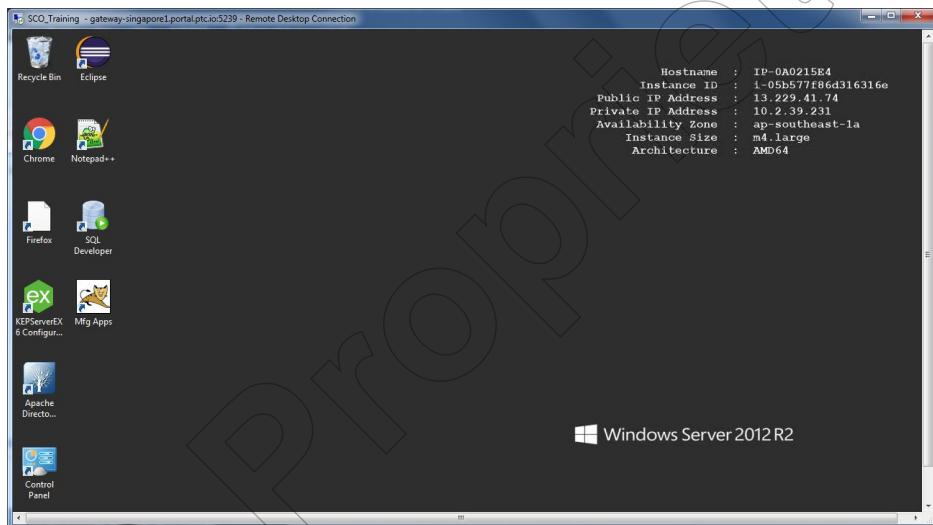
```

 Edit the value of 'screen mode id' settings to 2 if you want monitor size screen.

- Double-click the rdp. Copy the uniquely generated password from the Remote Access Credentials, refer step 10-B and paste in password field. Click **OK**.



13. To access the application using URL refer step 10-C and for application login details refer step 10-D.



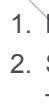
Task 2: Create VM Instance for Partner Training



Note that you need to create the VM instances in advance and share the VM ip address and ports with partner team. This is required if students will be accessing the VMs from their organization, organizations firewall will block the VM access and they need to open the ports for each instance.



Ideally you should create VM instances 10 days before the class and share the ip address and ports details with the partner team as they need to take multiple approvals to open the ports in their network.



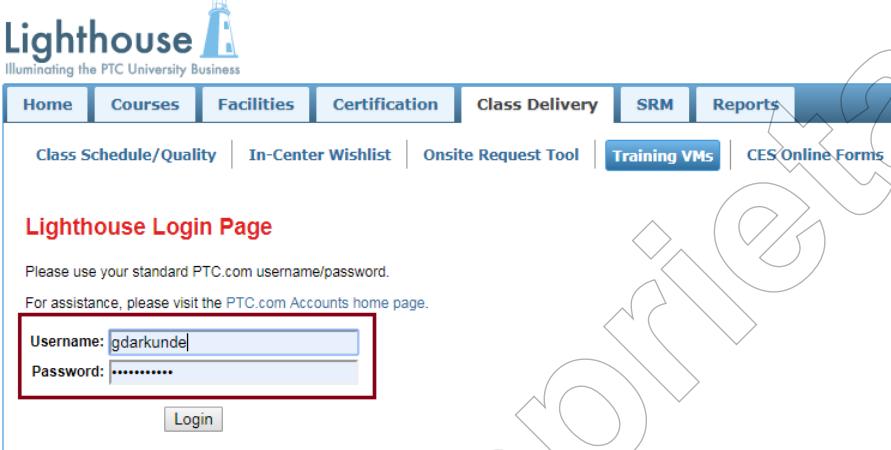
1. Browse [Technical Enablement Instructor Zone Page](#) and logon using your credentials.
2. Search your Course ID and note the VM template name mentioned in front of 'PTC Partners – Azure' under the VM Information column.

Example:

for course ID TE-00040, 'ThingWorx_Manufacturing_Apps_8.3_SCO-04' is the VM template name.

ILT/Virtual ILT Training Delivery Instructions							
Select the details link for the course (ID) you are scheduled to deliver to find all necessary information on how to prepare, deliver and assess the course.							
Course List							
#	Course/Assessment	Course ID	Format	Duration (Hours)	Details	Curriculum/s	VM Information
4	ThingWorx Manufacturing Apps – Custom App Development	TE-00040	Virtual ILT	8	Link	<ul style="list-style-type: none"> ThingWorx SCO C1 Curriculum for Developer 	<ul style="list-style-type: none"> PTC Internal – Cloud Portal: ThingWorx Manufacturing Apps 8.3 – Training VM-1 – Windows PTC Partners – Azure: ThingWorx_Manufacturing_Apps_8.3_SCO-02

3. Browse the [portal](#) and sign in using your PTC user account.



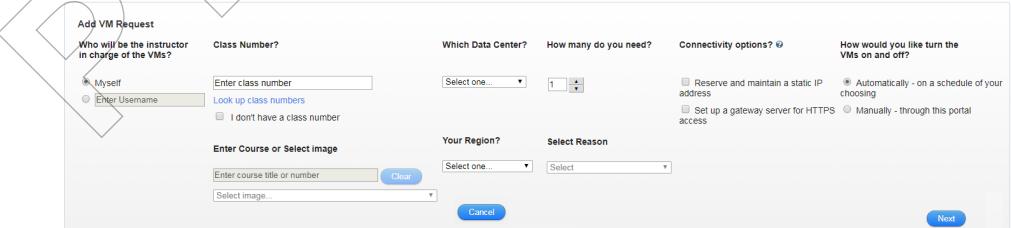
The screenshot shows the Lighthouse Login Page. At the top, there's a navigation bar with links: Home, Courses, Facilities, Certification, Class Delivery, SRM, Reports, Class Schedule/Quality, In-Center Wishlist, Onsite Request Tool, Training VMs (which is highlighted in blue), and CES Online Forms. Below the navigation, the text "Lighthouse Login Page" is displayed in red. It instructs users to use their standard PTC.com username/password and provides a link to the PTC.com Accounts home page. A red box highlights the login form fields: "Username: gdarkunde" and "Password:". A "Login" button is at the bottom of the form.

4. Click Add VM Request button.



The screenshot shows the VM Request Portal. The header features the PTC University Lighthouse logo. Below it, the text "VM Request Portal" and "VM Request Management for Ganeshkumar Darkunde". A red box highlights the "Add VM Request" button. To the right of the button, there's a note: "Not sure what to do? Download the job aid or step-by-step setup guide".

5. Add VM Request window opens.



The screenshot shows the "Add VM Request" window. It has several sections: "Who will be the instructor in charge of the VMs?", "Class Number?", "Which Data Center?", "How many do you need?", "Connectivity options?", and "How would you like to turn the VMs on and off?". Under "Who will be the instructor in charge of the VMs?", there are two radio buttons: "Myself" (selected) and "Enter Username". Under "Class Number?", there are three options: "Enter class number", "Look up class numbers", and "I don't have a class number" (selected). Under "Which Data Center?", there's a dropdown menu set to "Select one..." with a value of "1". Under "How many do you need?", there's a dropdown menu with values "1" and "4". Under "Connectivity options?", there are two checkboxes: "Reserve and maintain a static IP address" (unchecked) and "Set up a gateway server for HTTPS" (unchecked). Under "How would you like to turn the VMs on and off?", there are three checkboxes: "Automatically - on a schedule of your choosing" (unchecked), "Manually - through this portal access" (unchecked), and "Manually - through this portal access" (unchecked). At the bottom right, there's a "Next" button.

6. If you have class number, type it in **Class Number** field (will be share by training team). If you don't have class number, select **I don't have a class number** checkbox.
 7. Click the 'Select image...' drop-down menu under the Enter Course or Select image field and select the VM template noted in above step 2.

Class Number?

Enter class number

Look up class numbers

I don't have a class number

Enter Course or Select image

Enter course title or number Clear

ThingWorx_Manufacturing_Apps_8.3_SCO-02

8. Click the Select one drop-down menu under the Which Data Center? and select the nearest data center, (**EU, NA or AP**).
9. Click the Select one drop-down menu under the Your Region? and select your country.

Which Data Center?

West Europe (EU)

Your Region?

France

10. Select the **number of instances** you required from the 'How many do you need?' field.
11. Click the Select Reason drop down and select **Partner Customer Training**.

How many do you need?

6

Select Reason

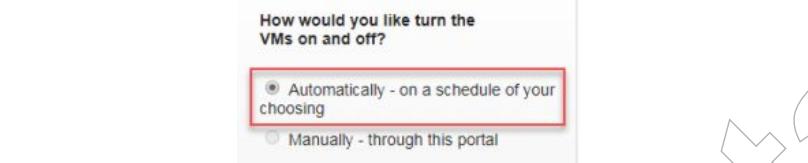
Partner Customer Training

12. In the Connectivity Options select **Reserve and maintain a static IP address** checkbox.

- A. If needed, ensure HTTPS options are checked.
 - Reserve and maintain a static IP address – Maintains same IP throughout the delivery time for all VMs
 - Set up a gateway server for HTTPS access – Allows alternate way to access VMs through browser



13. Keep default selection. Click **Next**.



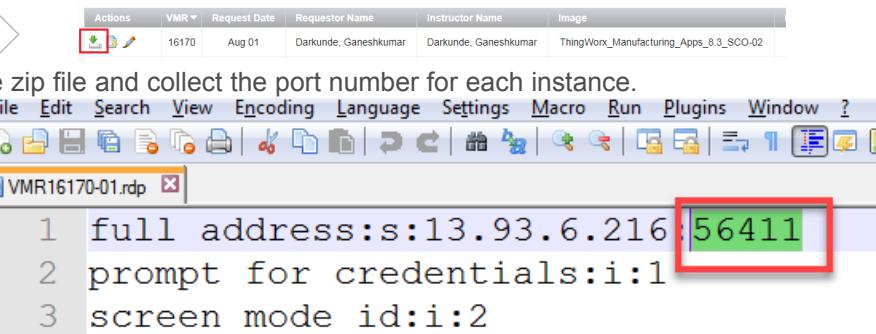
14. Select the class start date from 'What date do you want to CREATE the VMs in the request?' field.
 15. Select the class end date from the 'What date do you want to DELETE the VMs in the request?' field.



16. Select the VM start time 30 mins before the class.
 17. Select the VM stop time as per the class duration from What time do you want to STOP the VMs each day? Field.
 18. Lastly, select your time zone.



19. Click **Save**.
 20. If you didn't enter a class number, please enter a brief explanation of what the VMs will be used for. Click **Continue Saving**.
 21. After the approval, download the rdp files.



22. Extract the zip file and collect the port number for each instance.
 23. Share the port number with partner team to open the required ports. Once partners IT team opened the required ports, start the VMs for short duration and share the rdp file with partner

team to check if they are able to access the VMs. If partner team is able to access the VMs, you are set for the training.



This is required for the partner training, otherwise students will not be able to access the VMs.

24. To connect to your VM using the RDP file, simply double-click the RDP file and enter the VM credentials (username: PTCTrainingUser, password: stand1). Press **ENTER**.
25. Choose your desired language/country.
26. Open Google Chrome and type <http://localhost/Thingworx/>.
27. In the Sign in window, type Administrator in the username field and trUf6yuz2?_Gub in the password field.
28. If you are delivering training on Windchill, type <http://ptc-training.ptc.com/Windchill>. Use wcadmin:wcadmin as admin username and password.

This completes the exercise.

PTC Proprietary

PTC Proprietary

Module 2

Manufacturing Apps Overview

Module Overview

This module introduces you to the Manufacturing Apps. It also provides an overview of the ISA95 standard which is the basis of the manufacturing apps.

Objectives

After completing this module, you will be able to:

- Identify the IIoT (Industrial Internet of Things) use cases in manufacturing.
- Review the ThingWorx Manufacturing Apps.

Exercise 1: Exploring the ThingWorx Manufacturing Apps Features

Objectives

After successfully completing this exercise, you will be able to:

- Explore the features of ThingWorx Manufacturing Starter Apps.

Scenario

In this exercise, you explore the following ThingWorx Manufacturing Apps:

1. Asset Advisor
2. Controls Advisor
3. Production Advisor
4. Configuration and Setup

Task 1: Launch ThingWorx Manufacturing Apps.

1. Open the Chrome browser.
2. Type the following url: <http://localhost/Thingworx/FormLogin>Welcome>

 In case you face an error while launching the application, verify that the service Thingworx server is running.

3. Log on with the Administrator credentials.

Task 2: Explore Asset Advisor.

1. Click the **Asset Advisor** tile.

Task 3: Explore Controls Advisor.

1. From the ThingWorx Manufacturing Apps console, click the **Controls Advisor** tile.

Task 4: Explore Production Advisor.

1. From the ThingWorx Manufacturing Apps console, click the **Production Advisor** tile.

Task 5: Explore configuration and setup.

1. From the ThingWorx Manufacturing Apps console, click the **Configuration and Setup** tile.
2. Explore the following tabs: Equipment, Alerts, Notification Delivery, and Users.

This completes the exercise.

Module 4

Manufacturing Apps - Connectivity

Module Overview

This module talks about the connectivity in ThingWorx Manufacturing Apps. We will start with reviewing the connectivity technologies in ThingWorx platform. Then we will cover ThingWorx Industrial Connectivity and Edge Microserver to understand how they connect to the Manufacturing Apps. We will also understand how we can create a custom connection type for ThingWorx Manufacturing Apps.

Objectives

After completing this module, you will be able to:

- Review ThingWorx Connectivity technologies.
- Connect Manufacturing with ThingWorx Industrial Connectivity.
- Understand Manufacturing Apps' support for ThingWorx Edge MicroServer.
- Understand Manufacturing Apps' support beyond ThingWorx Industrial Connectivity.
- Connect Kepware to ThingWorx Manufacturing App

Exercise 1: Modeling Plant Assets in ThingWorx Industrial Connectivity

Objectives

After successfully completing this exercise, you will be able to:

- Create channel in ThingWorx Industrial Connectivity.
- Add device to ThingWorx Industrial Connectivity channel.
- Add tags to device in ThingWorx Industrial Connectivity.

Scenario

In this exercise, you take the role of ThingWorx Industrial Connectivity administrator and create the required channel. Also you will add required tags to the channels. Later you will use these channels in ThingWorx Manufacturing apps. For this training, we will use simulated values for the monitored properties.

Task 1: Create the new channel in ThingWorx Industrial Connectivity.

1. In the ThingWorx Industrial Connectivity window, right-click on **Connectivity** and select **New Channel**.
2. Select the **Simulator** driver from the drop-down list and click **Next**.
3. Specify the channel name, **LineGroup1**.
4. Keep the default settings in the **Add Channel Wizard** and finish creating the new channel.

Task 2: Create new devices to the channel.

1. Right-click **LineGroup1** channel and select **New Device**.
2. Type **SinkingEDM** in the Name field.
3. Keep the default settings in the **Add Device Wizard** and finish the creating the new device.
4. Follow the above steps and create three new devices named **WireEDM**, **GantryRobot**, and **WeldingRobot**.

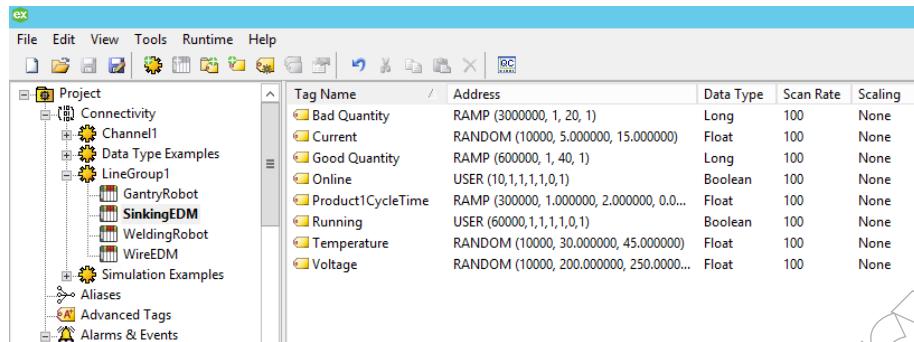
Task 3: Add tags to newly added devices.

1. Right-click **SinkingEDM** and select **New Tag**.
2. Type **Running** in the Name field.
3. Type **USER (60000,1,1,1,1,0,1)** in the Address field.

 This is a user-defined value. The syntax for user defined Boolean value is - USER (Rate, UserVal1, UserVal2, ..., UserValN). Refer to the KEPServerEx driver documentation for more details on the simulator functions.

4. Select **Boolean** as Data Type.
5. Click **OK** to finish the creating new tag.
6. Similarly, create the following tags for the same device.

Name	Address	Data Type	Scan Rate
Running	USER (6 0000,1,1,1,1,0,1)	Boolean	Default
Online	USER (6 0000,1,1,1,1,0,1)	Boolean	Default
Current	RANDOM (10000, 5, 15)	Float	Default
Temperature	RANDOM (10000, 30, 45)	Float	Default
Voltage	RANDOM (10000, 200, 250)	Float	Default
Good Quantity	RAMP (600000, 1, 40, 1)	Long	Default
Bad Quantity	RAMP (3000000, 1, 20, 1)	Long	Default
Product1CycleTime	RAMP (300000, 1,2,0.02)	Float	Default



This completes the exercise.

PTC Proprietary

Exercise 2: Configure KEPServerEX connectivity with ThingWorx Manufacturing Apps

Objectives

After successfully completing this exercise, you will be able to:

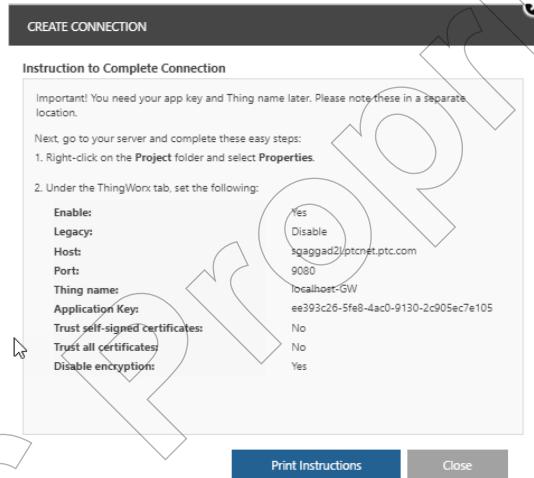
- Add the ThingWorx Industrial Connectivity server to ThingWorx Manufacturing Apps.
- Discover the devices in ThingWorx Manufacturing Apps.

Scenario

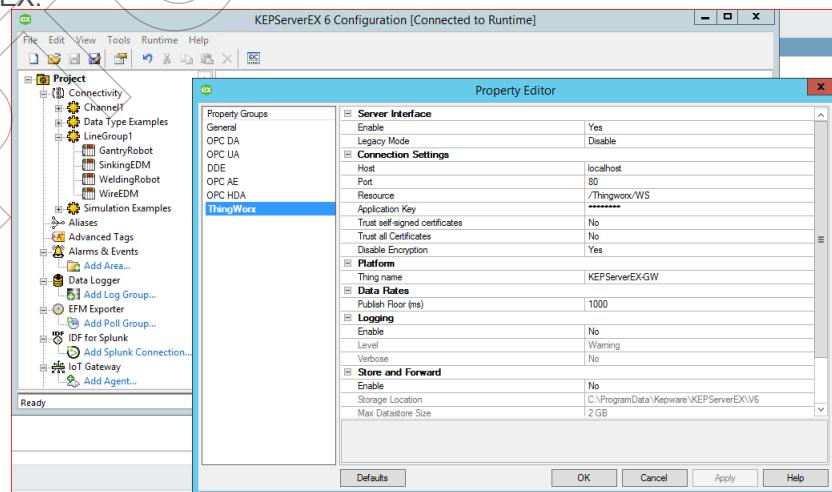
In this exercise, you take the role of administrator and configure the connectivity between ThingWorx Industrial Connectivity and ThingWorx Manufacturing apps.

Task 1: Add the server to ThingWorx Manufacturing Apps.

1. Log on to the ThingWorx Manufacturing Apps using administrator user.
2. Create a connection to ThingWorx Industrial Connectivity server.
3. Once the server is created, the application will prompt instructions to complete the connection.
Save this page.

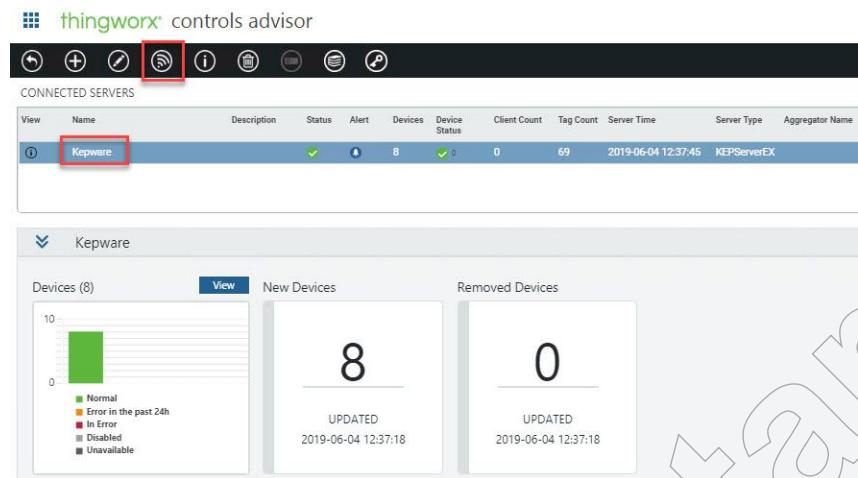


4. Follow the instructions from the above step and configure ThingWorx connection parameters in KEPServerEX.



Task 2: Discover the devices in ThingWorx Manufacturing Apps.

1. Select the added server and Click the Discover Devices icon at the top.



The screenshot shows the 'thingworx controls advisor' interface. At the top, there is a toolbar with various icons. One icon, which looks like a signal or a refresh symbol, is highlighted with a red box. Below the toolbar is a table titled 'CONNECTED SERVERS'. A row in this table is also highlighted with a red box and contains the name 'Kepware'. The table includes columns for View, Name, Description, Status, Alert, Devices, Device Status, Client Count, Tag Count, Server Time, Server Type, and Aggregator Name. The 'Devices' column for the Kepware server shows a value of 8. In the main content area, there is a section titled 'Kepware' with three sub-sections: 'Devices (8)', 'New Devices', and 'Removed Devices'. The 'Devices (8)' section includes a bar chart showing device status counts: Normal (green bar at 8), Error in the past 24h (orange bar at 0), In Error (red bar at 0), Disabled (grey bar at 0), and Unavailable (black bar at 0). The 'New Devices' section shows a count of 8 devices updated on 2019-06-04 12:37:18. The 'Removed Devices' section shows a count of 0 devices updated on 2019-06-04 12:37:18.

This completes the exercise.

PTC Proprietary

Exercise 3: Create the equipment structure in ThingWorx Manufacturing App

Objectives

After successfully completing this exercise, you will be able to:

- Create a new equipment structure in ThingWorx Manufacturing Apps.

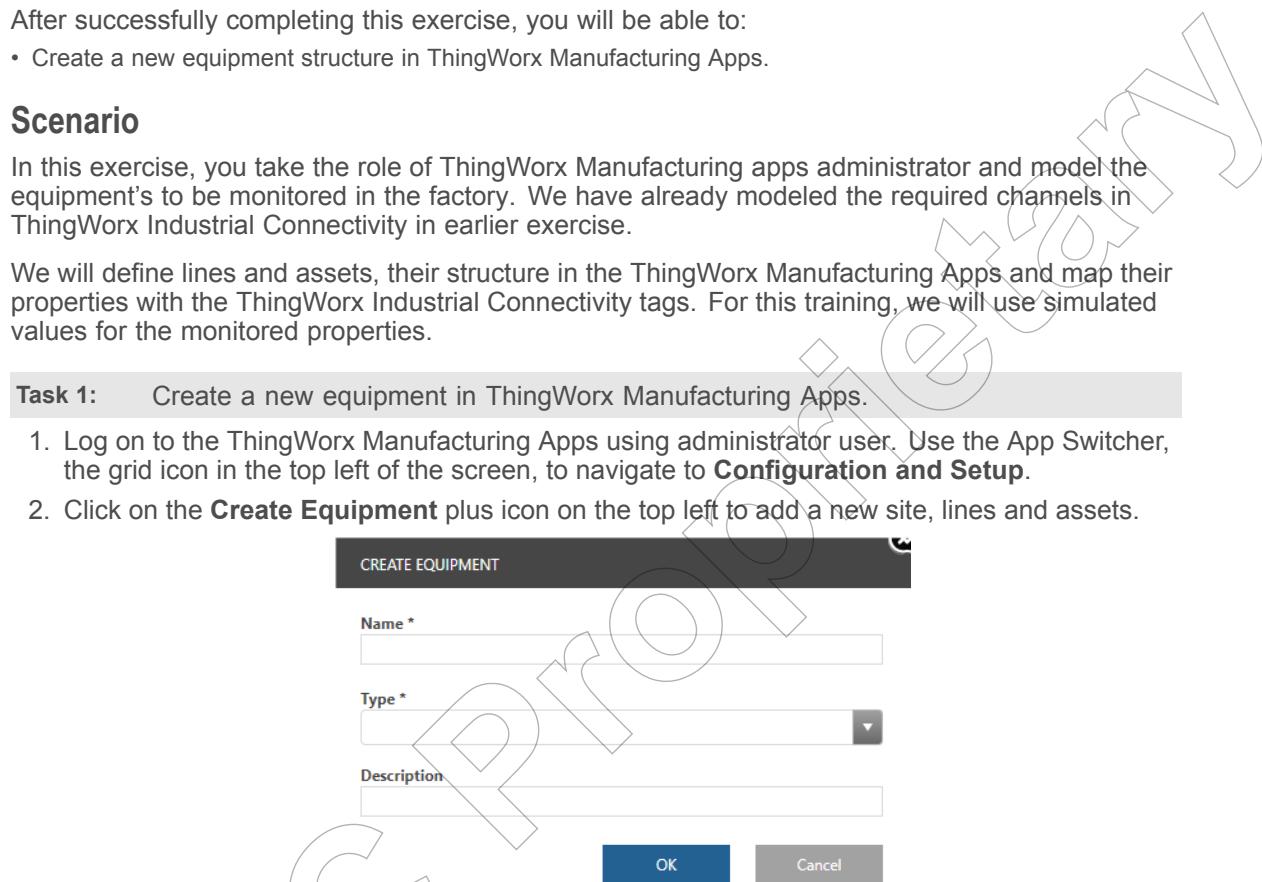
Scenario

In this exercise, you take the role of ThingWorx Manufacturing apps administrator and model the equipment's to be monitored in the factory. We have already modeled the required channels in ThingWorx Industrial Connectivity in earlier exercise.

We will define lines and assets, their structure in the ThingWorx Manufacturing Apps and map their properties with the ThingWorx Industrial Connectivity tags. For this training, we will use simulated values for the monitored properties.

Task 1: Create a new equipment in ThingWorx Manufacturing Apps.

1. Log on to the ThingWorx Manufacturing Apps using administrator user. Use the App Switcher, the grid icon in the top left of the screen, to navigate to **Configuration and Setup**.
2. Click on the **Create Equipment** plus icon on the top left to add a new site, lines and assets.



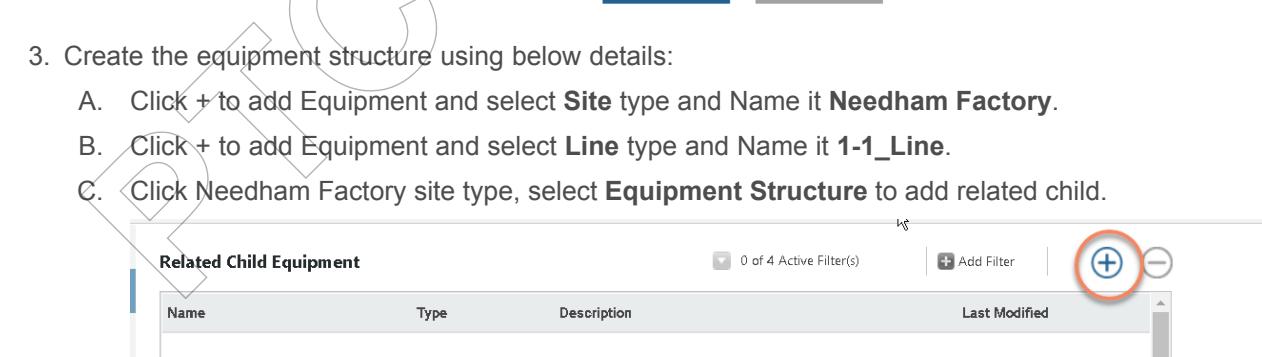
CREATE EQUIPMENT

Name *	<input type="text"/>
Type *	<input type="text"/>
Description	<input type="text"/>

OK **Cancel**

3. Create the equipment structure using below details:

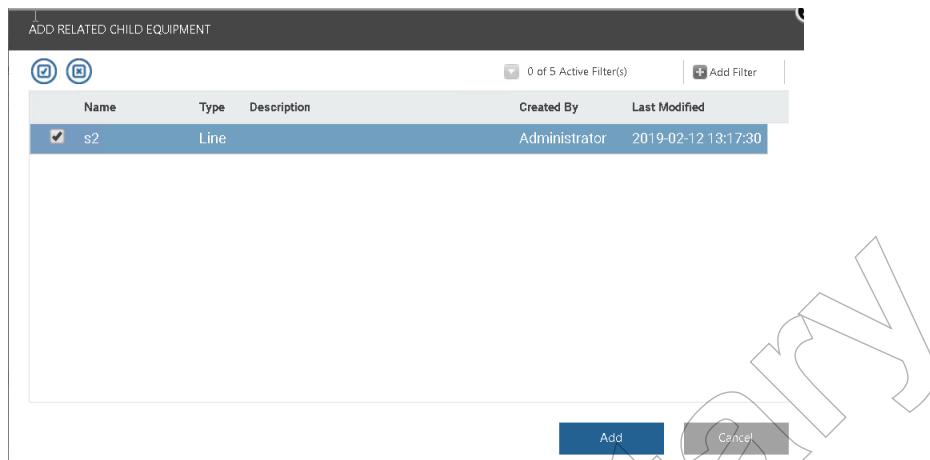
- A. Click + to add Equipment and select **Site** type and Name it **Needham Factory**.
- B. Click + to add Equipment and select **Line** type and Name it **1-1_Line**.
- C. Click Needham Factory site type, select **Equipment Structure** to add related child.



Related Child Equipment			
Name	Type	Description	Last Modified

0 of 4 Active Filter(s) Add Filter **(+)** **(-)**

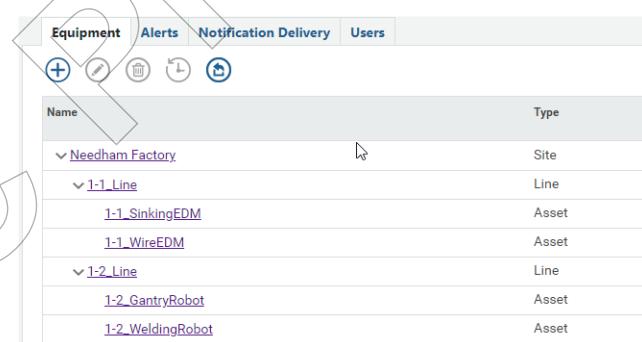
- D. The available lines Ex:**1-1_Line** should be displayed in the list.



- E. Similarly add two equipments of Asset type, **1-1_SinkingEDM** and **1-1_WireEDM**.
F. Now click the **1-1_Line**, click the Equipment Structure to add the related Assets.



- G. Create new equipment of Line type, **1-2_Line** and create a structure as shown in the image below.



 Define the equipment relationship using the **Equipment Structure** section in the **Configuration and Setup** app. Define child equipment for the relevant equipment. For example, in the image above, the Site has two children, **1-1_Line** and **1-2_Line**. Similarly, **1-1_Line** has two assets as child equipments.

This completes the exercise.

Exercise 4: Configure the Status Configuration Expressions and Performance Metrics for each Asset

Objectives

After successfully completing this exercise, you will be able to:

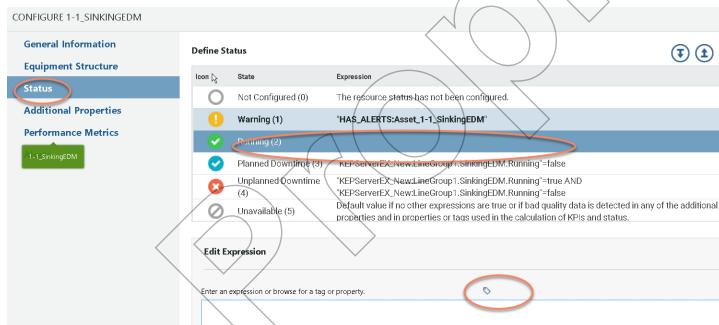
- Customizing the Default Status Configuration Expressions.
- Configuring the Performance Metrics.

Scenario

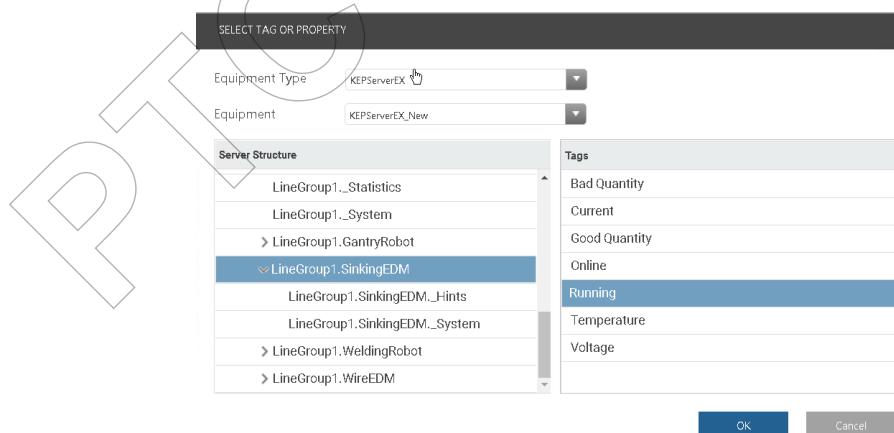
In this exercise, you will configure status configuration expressions for assets. Also you will configure performance metrics using ThingWorx Industrial Connectivity tags. Later we will take a look at values from the Production KPIs app.

Task 1: Configuring the Status Configuration Expressions for asset.

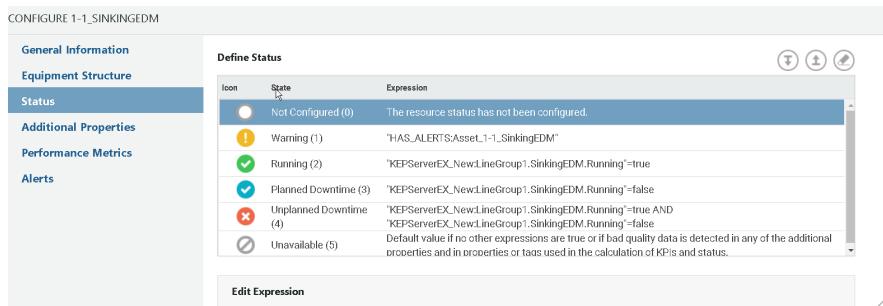
1. From Configuration and Setup app click on the 1-1_SinkingEDM asset.
2. Click the **Status** to define a status for the Running State. Ex: ["KEPServerEX_New:LineGroup1.SinkingEDM.Running"].
3. Click the **Browse for a tag or property** icon in the Edit Expression box.



4. Select the server and tags. Click **OK**.



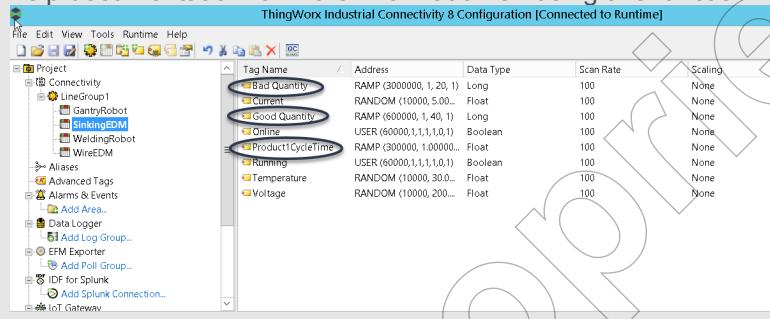
5. Click **Save**. Similarly add other Status expressions.



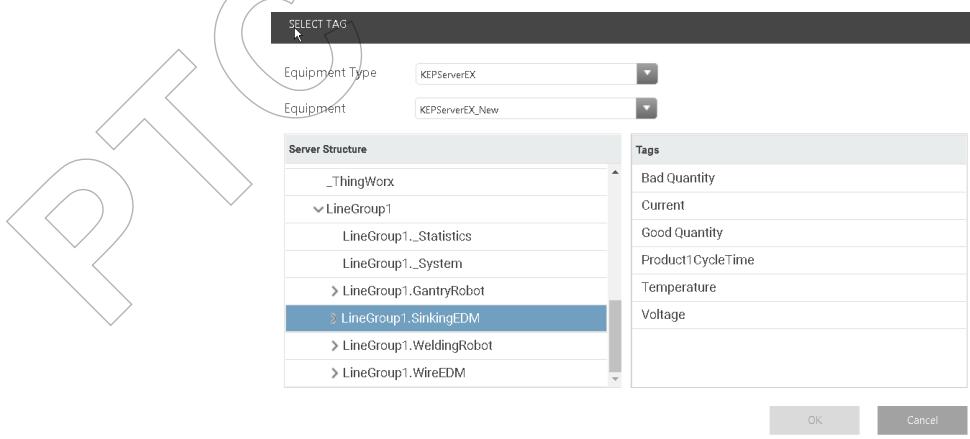
Task 2: Configuring the Performance Metrics.



In the sample application, we have defined the tags - **Good_Quality** and **Bad_Quality** and **Product1CycleTime** in **KEPServerEX**. These properties can be added in the Thingworx properties and remotely bind them. The **RAMP** function is used for these tags which increments the tag value through a numeric range. Refer to **KEPServerEx** driver help documentation for more information on using this function.



1. From Configuration and Setup app click the 1-1_SinkingEDM asset.
2. Click the Performance Metrics to define parameters for performance calculations.
3. Define the Good Count by clicking the **Browse for a tag** icon and choose the KepserverEx tags for **Good_Quality**.
4. Similarly, define Total Count using KepserverEx good and bad quantity tags.
5. Also, define the Ideal Runrate using KepserverEx Product1CycleTime tag.



6. Click **Save** once all expressions are defined.

Parameters for Performance Calculations

Enter an expression or browse for a tag.

The following operators are valid: +, -, *, /.

The following operands are valid: numbers and tags.

Strings between single quotes ('), KEPServerEX tags between double quotes (").

Good Count	34.0 Parts
'KEPServerEX_NewLineGroup1.SinkingEDM.Good Quantity'	
Total Count	49.0 Parts
'KEPServerEX_NewLineGroup1.SinkingEDM.Bad Quantity'+"KEPServerEX_NewLineGroup1.SinkingEDM.Good Quantity"	
Ideal Run Rate	0.9615384968074834 Part per Minute
1/"KEPServerEX_NewLineGroup1.SinkingEDM.ProductCycleTime"	

Save

KEPServerEX 6 Configuration [Connected to Runtime]

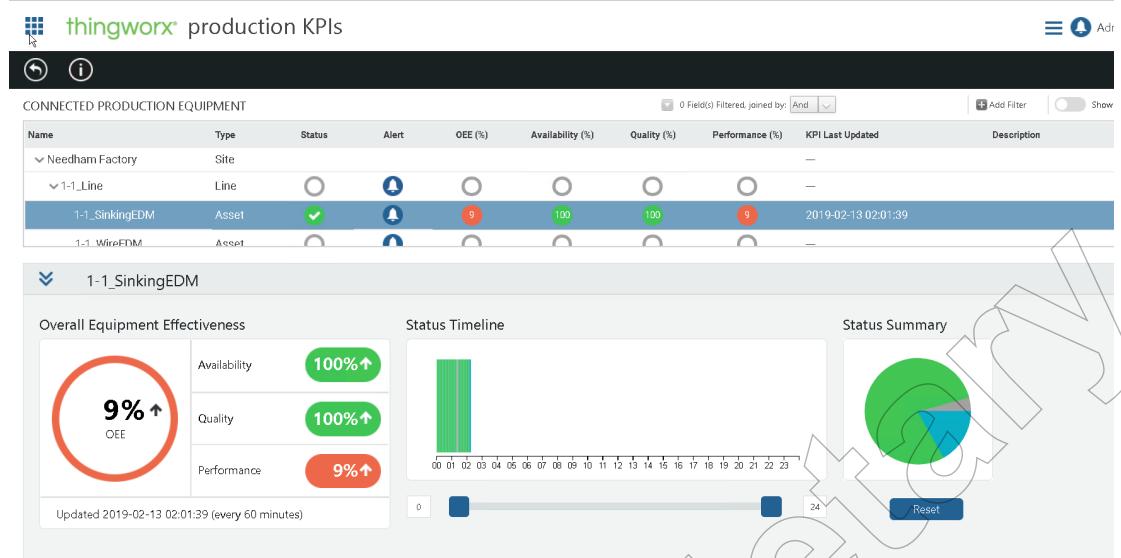
File Edit View Tools Runtime Help

Tag Name	/ Address	Data Type	Scan Rate	Scaling	Description
AbnormalCurrent_Co...	K3009.03	Boolean	100	None	
Bad Quantity	RAMP (300000, 1, 20, 1)	Word	100	None	
Current	K3003	Float	100	None	
Good Quantity	RAMP (600000, 1, 42, 1)	Word	100	None	
HighTemperature_Co...	K3009.02	Boolean	100	None	
LowVoltage_Control	K3009.01	Boolean	100	None	
Machines Online	K3009.05	Boolean	100	None	
Running	K3009.04	Boolean	100	None	
Temperature	K3007	Float	100	None	
Voltage	K3005	Float	100	None	

Ready Default User Clients: 5 Active tags: 463 of 463

Task 3: Test the application.

1. Click the Production KPI app.
2. Read the values against the SinkingEDM.



thingworx® production KPIs

CONNECTED PRODUCTION EQUIPMENT

Name	Type	Status	Alert	OEE (%)	Availability (%)	Quality (%)	Performance (%)
1-1_WireEDM	Asset	Green	Green	90	100	35	255
1-2_Line	Line	Grey	Green	90	100	35	255
1-2_GantryRobot	Asset	Green	Green	90	100	35	255
1-2_WeldingRobot	Asset	Green	Green	90	100	35	255



You may need to wait for a couple of minutes to reflect these values in the app.

This completes the exercise.



PTC Proprietary

Module 6

Customizing ThingWorx Manufacturing Apps

Module Overview

In this module, you learn how to customize the manufacturing apps. Primarily, the module explains how to model, connect, and build (UI) the manufacturing apps. The module focuses on the core capabilities to be used to customize the manufacturing apps.

Objectives

After completing this module, you will be able to:

- Understand the ThingWorx Development Process.
- Identify the entities to be customized to extend the ThingWorx Manufacturing Apps.
- Identify the building blocks provided by ThingWorx Manufacturing Apps for customization.
- Override the KPI calculations in ThingWorx Manufacturing Apps.
- Customize existing apps.
- Understand Status Expression Resource Provider.

Exercise 1: Customizing the Quality Calculations

Objectives

After successfully completing this exercise, you will be able to:

- Customize the quality calculation of equipment.

Scenario

Companies may calculate KPIs differently based on their business processes. Manufacturing apps provide a way to override existing KPI calculations. In this exercise, you override the existing quality calculations for the manufacturing apps. The VM contains sample data of good part count and bad part count in a mysql table. You use this data to calculate quality.

Task 1: Add the data for good part count and rejects.

- Open the database Thing PTC.MySQL.MfgAppQualityData in ThingWorx Composer.
- Execute the **AddSampleData** service with the following inputs:
 - date: use the current date with time 00:00:00, for example, 2018-08-09 00:00:00
 - badcount: any value between 1 and 3
 - shift: 1 (the value must be 1, 2, or 3)
 - goodcount: any value between 6 and 9
 - asset: select the thing for the asset, for example, Asset_1-1_WireEDM
- Execute the service.
- Similarly, add the sample data for other two shifts.

Task 2: Create services for getting the good count and bad count.

- In the PTC.MySQL.MfgAppQualityData Thing, create an SQL Query service with the following details:
 - Service name: GetGoodPartCount
 - Input parameters:
 - asset (String)
 - date (DateTime)
 - Output base type: Infotable
 - Datashape: Add a Data Shape, GoodQtyDS, with 1 field, GoodQty (Number)
- Add the following sql query in the script area:
`select GoodQty from mfgapptestdata where AssetId = [[asset]] and Date = [[date]];`
- Click **Done**.
- Similarly, add another SQL Query service, GetBadPartCount, for fetching the bad count number from the database. For this service, create a Data Shape, BadQtyDS with one field, BadQuantity (Number).
- Add SQL Query for this service.
`select BadQuantity from mfgapptestdata where AssetId = [[asset]] and Date = [[date]];`
- Save the thing.
- Test the services for the Thing we used in Task 1 (Asset_1-1_WireEDM).

Task 3: Overwrite the CustomizedQualityCalculation service.

-  In this task, we get the total of good part count and bad part count for the current date from the MySql database and use the values for the quality calculation.

- Open the Thing PTC.Factory.StatusExpressionResourceProvider.
- Override and edit the service **CustomizedQualityCalculation**.
- Type the following in the script area:

```
var date = Things["PTC.Factory.HelperServices"].GetCurrentDate();
```



PTC.Factory.HelperServices is pre loaded.

4. Select the **Entities** tab.
5. Search for the Thing PTC.MySQL.MfgAppQualityData.
6. Click the **Add** icon for the GetGoodPartCount service.
7. Edit the parameters for the service:

```
var params = {
  date: undefined /* DATETIME */,
  asset: undefined /* STRING */
};
to
var params = {
  date: date /* DATETIME */,
  asset: thingId /* STRING */
};
```

8. Edit the line:
9. Select the **Snippets** tab.
10. Select the **Infotable** category.
11. Click the **Add** icon for the Infotable for loop snippet.

12. Change the snippet code to:

```
var tableLength = goodcounts.rows.length;
var goodcount = 0;
var count = 0;
for (var x = 0; x < tableLength; x++) {
  var row = goodcounts.rows[x];
  count = row.GoodQty; // GoodQty is the field name of datashape GoodQtyDS
  goodcount = goodcount + count;
}
```



This code gives the total good count for the current date.

13. Similarly, add the following code to calculate the total bad part count for the current date:

```
var params = {
  date: date /* DATETIME */,
  asset: thingId /* STRING */
};
// result: INFOTABLE dataShape: BadQtyDS
var badcounts = Things["PTC.MySQL.MfgAppQualityData"].GetBadPartCount(params);
var tableLength = badcounts.rows.length;
var badcount = 0;
var count = 0;
for (var x = 0; x < tableLength; x++) {
  var row = badcounts.rows[x];
  count = row.BadQuantity; // BadQuantity is the field name of datashape BadQtyDS
  badcount = badcount + count;
}
```

14. Add the following code for quality calculations:

```
var total = goodcount + badcount;
var result = goodcount / total;
```

15. Validate the following code:

```
var date = Things["PTC.Factory.HelperServices"].GetCurrentDate();
// Get total good count of parts for the current date
var params = {
  date: date /* DATETIME */,
  asset: thingId /* STRING */
};
// result: INFOTABLE
var goodcounts =
  Things["PTC.MySQL.MfgAppQualityData"].GetGoodPartCount(params);
var tableLength = goodcounts.rows.length;
```

```

var goodcount = 0;
var count = 0;
for (var x = 0; x < tableLength; x++) {
var row = goodcounts.rows[x];
count = row.GoodQty; // GoodQty is the field name of dataspace GoodQtyDS
goodcount = goodcount + count;
}
// Get total bad count of parts for the current date
var params = {
date: date /* DATETIME */,
asset: thingId /* STRING */
};
// result: INFOTABLE dataspace: BadQtyDS
var badcounts = Things["PTC.MySQL.MfgAppQualityData"].GetBadPartCount (params);
var tableLength = badcounts.rows.length;
var badcount = 0;
var count = 0;
for (var x = 0; x < tableLength; x++) {
var row = badcounts.rows[x];
count = row.BadQuantity; // BadQuantity is the field name of dataspace BadQtyDS
badcount = badcount + count;
}
////calculate quality
var total = goodcount + badcount;
var result = goodcount / total;

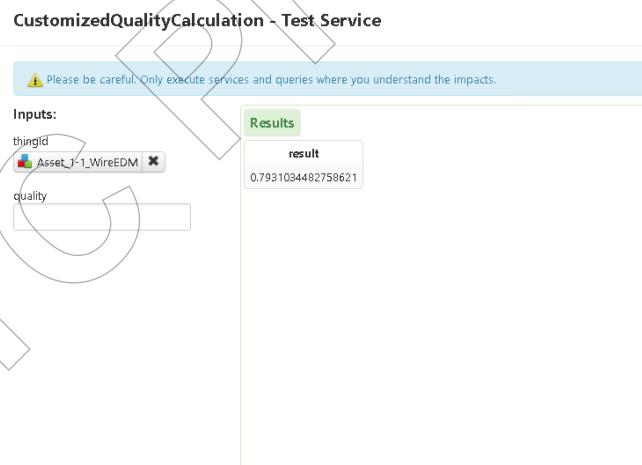
```

16. Click **Done**.

17. Save the Thing.

Task 4: Test the CustomizedQualityCalculation service.

1. Click the **Execute** button for the CustomizedQualityCalculation service.
2. Specify Asset_1-1_WireEDM as the thingId.
3. Execute the service.
4. You should view the calculated quality value in the result.



This completes the exercise.

Exercise 2: Visualizing the Customized Quality Calculations in ThingWorx Manufacturing Apps

Objectives

After successfully completing this exercise, you will be able to:

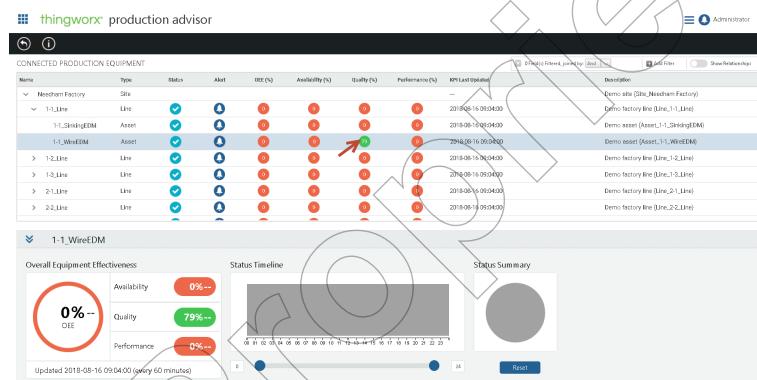
- Visualize the customized quality calculations of an equipment.

Scenario

In this exercise, you visualize the customized quality calculations in Production KPI.

Task 1: Visualize the quality data in Production KPI.

1. Open the Production KPIs app.
2. Notice that the customized quality data is reflected for the asset we used.



3. Notice that the OEE is also calculated based on the customized quality value.



OEE = Availability X Quality X performance

This completes the exercise.

Exercise 3: Customizing Controls Advisor

Objectives

After successfully completing this exercise, you will be able to:

- Customize the existing out-of-the-box Controls Advisor.

Scenario

In this exercise, we will customize the mashup for the device list page, which launches when you click the View button or the Device Status link in Controls Advisor.

Task 1: Open PTC.Factory.KEPServerEX.C_DeviceList_8.3.0_05.

- Open ThingWorx Composer.
- In the spotlight search field, search for the mashup PTC.Factory.KEPServerEX.C_DeviceList_[ReleaseVersion].



In the VM provided for this exercise, the name of this thing will be PTC.Factory.KEPServerEX.C_DeviceList_8.3.0_05.

Task 2: Edit the gridadvanced-DeviceList widget.

- Go to the Workspace tab.
- Under Filter Workspace, search for the widget gridadvanced-DeviceList.
- Select the widget under search.
- Click the **Configure Grid Columns** icon under the widget properties.

- In the Configure Widget window, clear the check boxes under the SHOW column except for the Device Name, Channel, and Device ID columns displayed in the grid.

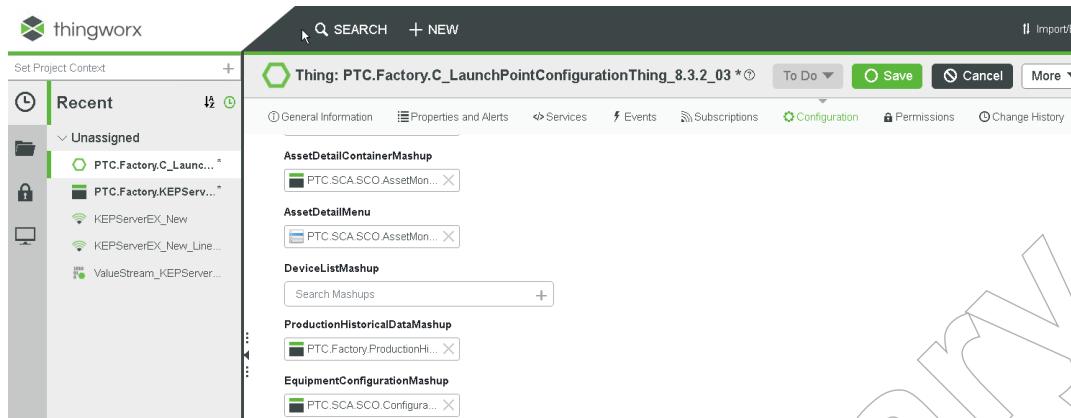
Configure Widget

	Deselect All	Select All	DRAG TO REORDER	EXCLUDE	SHOW
DeviceName	<input type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
Channel	<input type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
DeviceId	<input type="checkbox"/>	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>
inErrorHTMLString	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
AlertIcon	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
_SecondsInError	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
ConnectionType	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
_Simulated	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
_AutoDemoted	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
_PendingReads	<input type="checkbox"/>	<input checked="" type="checkbox"/>			<input type="checkbox"/>

- Save the mashup.

Task 3: Configure the Launch Point.

- Open PTC.Factory.KEPServerEX.C_DeviceList_8.3.0_05 in ThingWorx Composer.
- In Configuration, change the value for DeviceListMashup to the name of the customized mashup edited in the previous step.



3. Go to **Properties > Alerts** and ensure that the Enabled property is set to true. This is the default value.
4. Save the configuration thing.

Task 4: Verify the mashup.

1. Open the PTC.FactoryConsole mashup.
2. Open Controls Advisor.
3. Click the **View** button.
4. The Device List table should only list Device Name, Channel, and Device ID columns.

Device Name	Channel	Device ID
1-1_SinkingEDM	LineGroup1	1
1-1_WireEDM	LineGroup1	1
1-2_GantryRobot	LineGroup1	1
1-2_WeldingRobo	LineGroup1	1
1-3_CNCMill	LineGroup1	1
1-3_CNCTurning	LineGroup1	1
2-1_Bender	LineGroup2	1
2-1_LaserCutter	LineGroup2	1
2-2_Grinder	LineGroup2	1
2-2_Press	LineGroup2	1
3DPrinter	LineGroup2	1
WaterCutter	LineGroup2	1
Gen	Sim	1
Gen1	Sim	1
Gen2	Sim	1

This completes the exercise.

PTC Proprietary

Module 7

Other Customization Entities

Module Overview

This module explains the additional entities to customize the Manufacturing apps. In this module, you create a new app by using some of the Manufacturing Apps entities discussed in this course.

Objectives

After completing this module, you will be able to:

- Identify the entities for customizing UI components of the ThingWorx Manufacturing apps.
- Learn about Equipment Structure Relationship Rules.
- Create a new Manufacturing App.

Exercise 1: Writing Custom Data Shapes and Services

Objectives

After successfully completing this exercise, you will be able to:

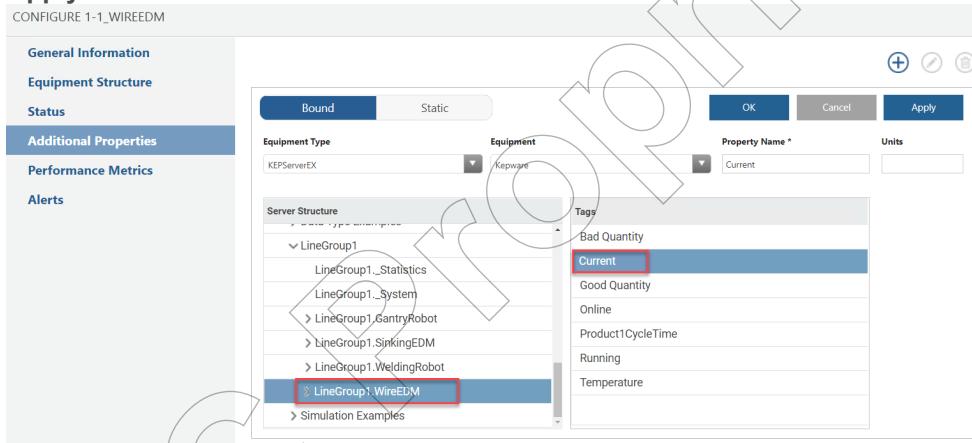
- Write custom services which will be used later for creating a custom manufacturing app.

Scenario

In this exercise, we assume that you want to create a new customized mashup to display custom data. For this, we will create a few custom Data Shapes and services which will be used in later exercises to create custom mashup.

Task 1: Additional asset properties using ThingWorx Industrial Connectivity.

1. Logon to the ThingWorx Manufacturing Apps and open **Configuration and Setup** app.
2. Click **1-1_WireEDM**.
3. Click **Additional Properties**.
4. Click **New Property** icon to add new property.
5. Select the Equipment Type and Equipment.
6. Select **WireEDM** from the Server Structure column and select **Current** from the Tag column. Click **Apply**.



7. Similarly create Temperature and Voltage properties.

Task 2: Create MFG_AssetProperty_DS Data Shape.

1. Open the New Thingworx Composer.
2. Type Data in the +New from Navigation Bar.
3. Select **Data Shape**.
4. Type the name as **MFG_AssetProperty_DS**.
5. Go to the Field Definitions section.
6. Create the following fields.

Field Name	Base Type
Current	Number
Voltage	Number
Temperature	Number

7. Click the **Done** button.
8. **Save** the Data Shape.

Task 3: Create the GetAssetDataTableFromFilter service.

1. Type **PTC.SCA.SCO.AssetMonitor** in the spotlight search field.

2. Select **PTC.SCA.SCO.AssetMonitor.AssetList.AssetListServiceController**.

3. Click the **Services** link.

4. Click the **Add My Service** button.

5. Create a new service in it with the following attributes:

A. Service Name: GetAssetDataTableFromFilter

B. Inputs parameter: NONE

C. Output parameter:

Name	BaseType	Data Shape	Infotable Type
Result	INFOTABLE	PTC.SCA.SCO.AssetMonitor.AssetList.AssetIdentityDataShape	Just Infotable

6. In the Script section, type the following code:

```
var params = {
pageNumber: undefined /* INTEGER */,
pageSize: 100 /* INTEGER */,
filters: undefined /* JSON */
};
var FinalInfo = me.GetFilteredAssetList(params);
result=FinalInfo.Data;
```

7. Click the **Done** button.

8. Save the Thing PTC.SCA.SCO.AssetMonitor.AssetList.AssetListServiceController.

9. Test your service.

GetAssetDataTableFromFilter - Test Service

Inputs:
No inputs

Results

name	displayName	image	attribut
Asset_1-1_SinkingEDM	1-1_SinkingEDM		
Asset_1-1_WireEDM	1-1_WireEDM		

Execute Service **Create DataShape from Result** **Close**

Task 4: Create the GetAssetProperty service.

1. Click the **Add My Service** button.

2. Create a new service in it with the following attributes:

A. Service Name: GetAssetProperty

B. Inputs parameter:

Parameter Name	BaseType
displayId	STRING

C. Output parameter:

Name	BaseType	Data Shape	Infotable Type
Result	INFOTABLE	MFG_AssetProperty_DS	Just Infotable

3. In the Script section, type the following code:

```
var params = {
displayId: displayId /* STRING */
};
// result: INFOTABLE dataShape: PTC.Factory.PhysicalAssetPropertyDataShape
var infotable =
```

```
Things["PTC.Factory.PlantStatusUtils"].GetExtraPhysicalAssetPropertiesAndValue  
sForBinding(params);  
var params = {  
    infoTableName : "InfoTable",  
    dataShapeName : "MFG_AssetProperty_DS"  
};  
// CreateInfoTableFromDataShape(infoTableName:STRING("InfoTable"),  
dataShapeName:STRING):INFOTABLE(MFG_AssetProperty_DS)  
var result =  
Resources["InfoTableFunctions"].CreateInfoTableFromDataShape(params);  
// MFG_AssetProperty_DS entry object  
var newEntry = new Object();  
newEntry.Temperature = infotable.rows[1].value; // NUMBER  
newEntry.Voltage = infotable.rows[2].value; // NUMBER  
newEntry.Current = infotable.rows[0].value; // NUMBER  
result.addRow(newEntry);
```

4. Click the **Done** button.
5. Save the Thing PTC.SCA.SCO.AssetMonitor.AssetList.AssetListServiceController.

This completes the exercise.

PTC Proprietary

Exercise 2: Creating a New Custom Manufacturing App

Objectives

After successfully completing this exercise, you will be able to:

- Add a custom app on the default application console page of Manufacturing app, that is, PTC.FactoryConsole.

Scenario

In this exercise, you will add a custom app to the default application console page of Manufacturing App, which when clicked will launch a custom mashup.

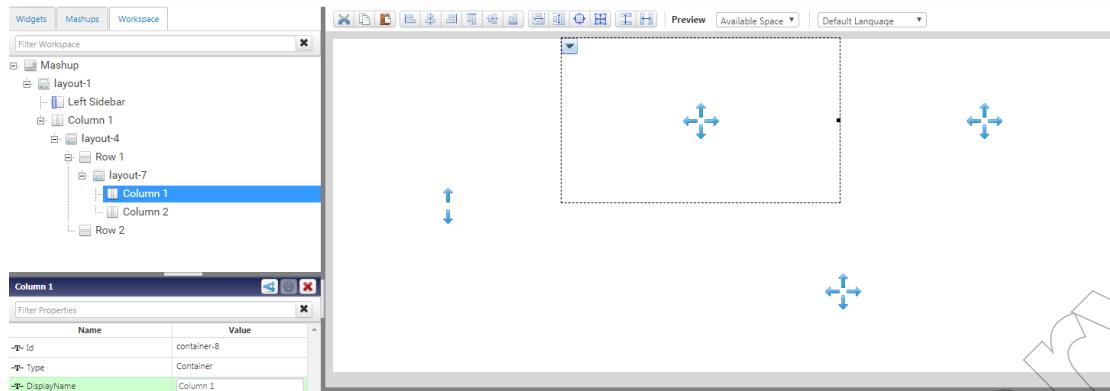
For this custom mashup, we will use custom data shapes and services that we created in the previous exercise.

Task 1: Create a responsive mashup.

1. Open the New ThingWorx Composer.
2. Create new mashup with default settings.

Task 2: Define the layout area.

1. Type **MyTestAppMashup** in the Name field.
2. Select the tags **sca-common** and **sca-mfg**.
3. Click the **Save** button.
4. Click the **Design** tab.
5. Type **La** in the Filter Widgets field on the Widgets panel.
6. Select **Layout**.
7. Drag the Layout widget from the upper-left panel to the upper-central panel.
8. Ensure Horizontal is selected and then select the **Left Sidebar** check box.
9. Select **1 Column** from the Columns drop-down list.
10. Click the **Done** button.
11. Select the **Workspace** tab.
12. Verify that **Layout** is selected on the Workspace panel.
13. Click the **Spacing** drop-down list for the layout.
14. Select **none**.
15. Type **300** in the LeftSidebarWidth property field and press TAB.
16. Select the **Widgets** tab.
17. Verify that La is still populated in the filter field of the Widgets panel and drag Layout to the Column1 section.
18. Select **Vertical**.
19. Validate that the two rows, no header, and no footer are selected and click **Done**.
20. Click the right row in the upper-right panel, that is, **Row1**, in the Workspace panel.
21. Select the **Widgets** tab. Verify that La is still populated in the filter field of the Widgets panel and drag Layout to the upper row, that is, the Row1 section.
22. Validate that the two columns are selected and click **Done**.
23. Your final mashup layout should look like the following:

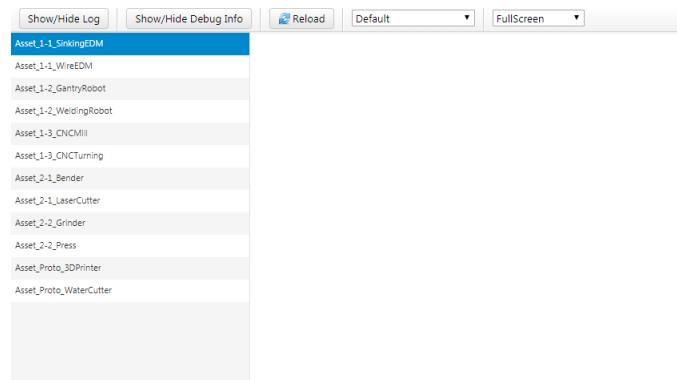


Task 3: Add a list of all assets in the layout.

1. Select the **Widgets** tab.
2. Type **Li** in the filter field on the Widgets panel.
3. Select the **List** widget.
4. Drag the List widget to the Left Sidebar.
5. Verify that the Data tab is selected in the upper-right panel and click the **Add entity +** icon.
6. Click the **Select Entities** field and type **PTC.SCA.SCO.AssetMonitor.AssetList.AssetListServiceController**.
7. Select the above thing.
8. In the Select Services filter, type **GetAsset**.
9. Click the **Add ➔** icon for the **GetAssetDataTableFromFilter** service.
10. Select the **Mashup Loaded?** check-box.
11. Click the **Add ➔** icon for the **GetAssetProperty** service.
12. Click the **Done** button.
13. Select the **GetAssetDataTableFromFilter** service.
14. Click the **Expand +** icon next to All Data in the Data panel.
15. Select the **All Data** row of Returned Data.
16. Drag the All Data row to the List widget.
17. Select **Data**.
18. In the Properties section of the List widget, select **name** from the drop-down list of the DisplayField and ValueField properties.
19. Select **AutoSelectFirstRow** check box.

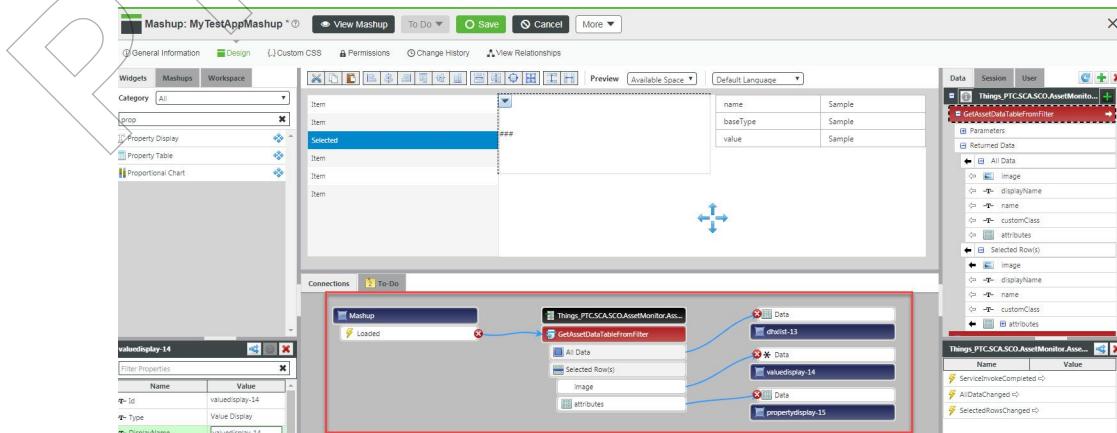
Task 4: Save and test the mashup.

1. Click the **Save** button.
2. Click the **View Mashup** button.



Task 5: Add the Value and Property Display widgets.

1. Edit the mashup again.
2. Select the central column, that is, **Column1**.
3. Select the **Widgets** tab.
4. Type **value**.
5. Select the **Value Display** widget.
6. Drag the Value Display widget to the central column of the layout.
7. Select the **Data** tab in the upper-right panel.
8. Select the **GetAssetDataTableFromFilter** service.
9. Click the **Expand**  icon next to Selected Row(s) of the GetAssetDataTableFromFilter service.
10. Drag the image from the Data panel to the central column layout.
11. Select **Data**.
12. Select the top-right column, that is, **Column 2**.
13. Select the **Widgets** tab.
14. Type **proper**.
15. Select the **Property Display** widget.
16. Drag the Property Display widget to the top-right column, that is, Column 2 of the layout.
17. Select the **GetAssetDataTableFromFilter** service.
18. Click the **Expand**  icon next to Selected Row(s) in the Data panel.
19. Drag the attributes row to the Property Display widget.
20. Select **Data**.



Task 6: Add the Progress Bar widget.

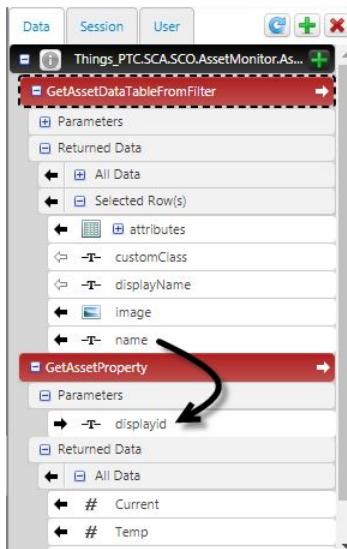


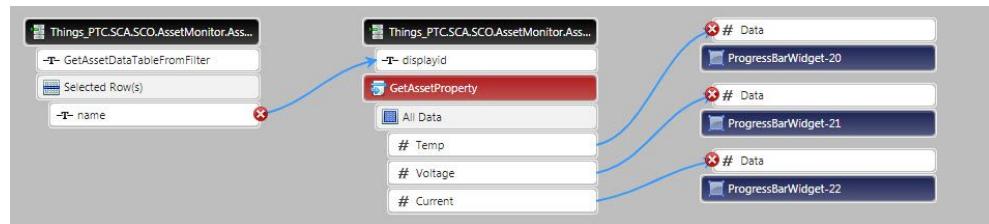
If you are using a PTCU VM the widget name may be ProgressGauge.

1. Edit the mashup again.
2. Select the bottom row, that is, **Row 2**.
3. Select the **Widgets** tab.
4. Search for the Layout widget. Drag the widget on the bottom row.
5. Select the three columns.
6. Click **Done**.
7. Type **prog** in the filter field on the Widgets panel.
8. Select **ProgressBarWidget**.
9. Drag ProgressBarWidget in all the three columns of the bottom row, that is, **Row 2**.
10. Select the **GetAssetProperty** service in the Data tab.
11. Click the **Expand**  icon next to All Data in the Data rows.
12. Drag Temp into the first ProgressBarWidget.
13. Select **Data**.
14. Similarly drag Voltage and Current in the second and third ProgressBarWidget, respectively.
15. Select **Data**.

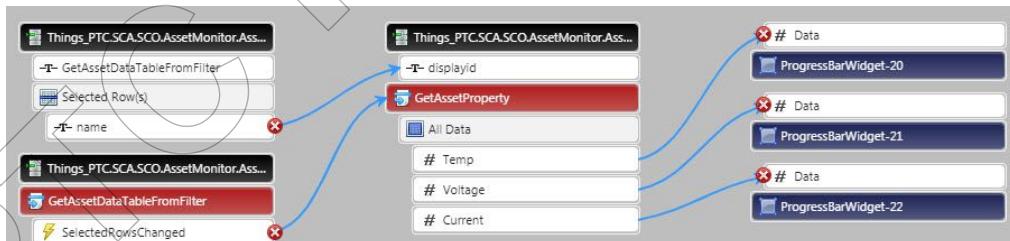
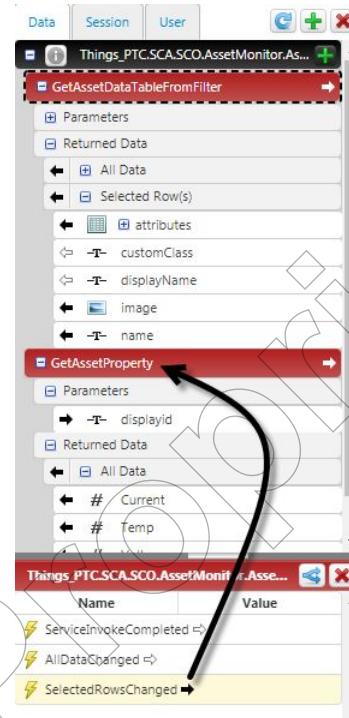


16. Select the **GetAssetDataTableFromFilter** service in the Data panel.
17. Expand Selected Row(s) under Returned Data.
18. Select the **GetAssetProperty** service in the Data panel.
19. Expand the Parameters section.
20. Drag the **name** attribute from the GetAssetDataTableFromFilter service to **displayId** input parameter on the GetAssetProperty service.





21. Select the **GetAssetDataTableFromFilter** service.
22. Drag its **SelectedRowsChanged** event to the **GetAssetProperty** service.



23. Save the mashup.

Task 7: Create new localization tokens.

1. In the left browser panel, click **SYSTEM > Localization Tables**.
2. Select the Default localization table.
3. Go to the Localization Tokens section.
4. Click the **Add** button.
5. In the New Localization Token window, create two new tokens with the following parameters:

Name	Token Value
PTC.CustomApp.MyAppTitle	My CustomApp
PTC.CustomApp.MyAppLabel	My CustomApp Viewer

6. Save the Localization table.

Task 8: Import the custom master mashup.

1. Go to ThingWorx Composer.
2. Go to Import From File under the Import/Export option.
3. Select the **Use Default Persistence Provider** check box.
4. Click the **Browse** Button.
5. Browse to the **Mashups_MyCustomMasterMashup.xml** file location (that is, D:\Student\LabFiles).
6. Import the file.
7. Open Mashup MyCustomMasterMashup.
8. Verify that its property mashupTitle is set to [[PTC.SCA.Common.WindowTitle]].
9. Open the mashup MyTestAppMashup.
10. Set the property Master value as MyCustomMasterMashup.
11. **Save** MyTestAppMashup.

Task 9: Add a new tile to the main application console.

1. Click **+New** and type **me**. Select **Media**.
2. Type the name as **MyCustomAsset**.
3. Click the **Change** button for image.
4. Browse to the MyCustomAsset.jpg file location (that is, D:\Student\LabFiles\Tile Icons).
5. Click the **Change** button.
6. **Save** the media entity.
7. Search for the Thing
PTC.FactoryConsole.C_DataTable_[ReleaseVersion]. (In the VM provided for this exercise, the name of this thing will be PTC.FactoryConsole.C_DataTable_8.3.0_05.)
8. Click **Mashup**.
9. In the Entries Editor section, edit the following parameters:

Parameters	Value
Tile Order	40
TileMashupOrURL	MyTestAppMashup
TileIcon	MyCustomAsset
TileBackground	PTC.Yellow
TileLabel	[[PTC.CustomApp.MyAppTitle]]
TileTooltip	My Custom App



Note that for each Tile Order, unique key is generated. You can also change other parameters as per your requirement.

10. Click the **Add/update** button.
11. Save the Thing PTC.FactoryConsole.C_DataTable_8.3.0_05.
12. Open PTC.Factory.C_LaunchPointConfigurationThing_8.3.0_05.
13. Click **Configuration**.
14. Ensure that the TilesDataTable launch point value in PTC.Factory.C_LaunchPointConfigurationThing_[ReleaseVersion] points to PTC.FactoryConsole.C_DataTable_[ReleaseVersion], that is, Thing PTC.FactoryConsole.C_DataTable_8.3.0_05.
15. Launch the main application console mashup, that is, PTC.FactoryConsole.
16. This should list our custom app tile.



17.

Result:

When you click the custom tile, it should take you to our custom Manufacturing app, that is, MyTestAppMashup.



This completes the exercise.

Exercise 3: Creating Custom Roles

Objectives

After successfully completing this exercise, you will be able to:

- Create custom roles.
- Provide access to the custom roles.
- Create new user.

Scenario

You can define your own roles and assign permissions to those roles in addition to, or instead of, the roles provided with the ThingWorx Apps. This enables you to tailor the application to your business processes.

In this exercise, you create a custom role for Plant Manager. The role should have the accesses of Maintenance Manager and Production Manager.

Task 1: Create a new role in Composer.

1. In ThingWorx Composer, click **User Groups** under Security.
2. Click **New** to create a new user group.
3. In the Name field, type **Plant Manager**.
4. Select the following tags:
 - A. PTC: factory-mv
 - B. Role: Factory-UserGroup
5. Save the user group.
6. Select **FactoryUsers** from the list of User Groups and click **Edit Members**.
7. Select the **Plant Manager** user group from the list on the left and move it to the list on the right.
8. Click **Save** to save the changes and close the window.
9. Save the FactoryUsers user group.

Task 2: Assign access rights to the custom role.

1. In ThingWorx Composer, click **User Groups** under Security.
2. Click the **Maintenance Manager** user groups to apply the same access rights to your custom role.
3. Click **Edit Members**.
4. Select the **Plant Manager** role from the list on the left, and move it to the list on the right.
5. Save the changes to the user group.
6. Repeat these steps to apply the rights of the Production Manager group to the Plant Manager role.

Task 3: Create a new user in the Manufacturing App with the new role.

1. From the ThingWorx Manufacturing Apps console, open Configuration and Setup.
2. On the Users tab, click **+** to create a new user. Observe that the Plant Manager role appears in the Role Assignment panel.
3. Fill the required details on the Create User pop-up window, and select the **Plant Manager** role.

CREATE USER

User	Security
User Name * jdoe	Password * *****
First Name * John	Confirm Password * *****
Last Name * Doe	
Language English	
Contact Information	
Email * jdoe@ptc.com	
Work Phone * 26985632	
Mobile Phone	
Notification Preference	
<input checked="" type="checkbox"/> Email <input type="checkbox"/> SMS <small>Standard message rates may apply.</small>	
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

4. Click **OK**.
5. Open ThingWorx Manufacturing Apps in a private window.
6. Log on with the new user created.
7. Observe that the user can access the apps for the roles: Maintenance Manager and Production Manager.

This completes the exercise.