

# ThingWorx Tips and Tricks

PTC University

One of the best ways to understand how to use ThingWorx is to observe it in action. These five tips are examples of the types of hands-on demonstrations our instructor-led courses provide to speed your learning curve.

## Table of Contents

- Tip 1: Leverage ThingWorx connectivity to consume data from existing web services within an a ThingWorx application
  - Create a Data Shape
  - Create a new Service on a Thing
  - Create a new Mashup
- Tip 2: Use tags to group and more easily find related items
- Tip 3: Establish naming conventions
- Tip 4: Set pass through permissions to limit the use of a service
- Tip 5: More easily create a new thing by determining how an existing thing was created

ThingWorx is PTC's platform that enables developers to deliver end-to-end Internet of Things solutions.

The ThingWorx IoT Platform provides a complete application design, runtime, and intelligence platform. ThingWorx's model-based design and search-based intelligence enable companies to reduce application development efforts by 10X, allowing them to minimize cost, risk and time to market. The ThingWorx platform combines the key functions of Web 2.0, search, and social collaboration and applies it to the world of "things", including connected products, machines, sensors, and industrial equipment.

ThingWorx is an incredibly open and flexible system. To fully harness its power, developers need to understand what it can do. One of the best ways to learn about ThingWorx's many capabilities is to see them in action making something productive. That's because ThingWorx makes it easy to leverage existing services or logic to eliminate the need to create everything from scratch. PTC instructor-led training gives you hands-on demonstrations of key ThingWorx features. You're guided by the most knowledgeable experts who help you learn by doing rather than having to struggle to teach yourself using a manual. The following tip gives you a step-by-step example of how you can quickly build a powerful service in ThingWorx.

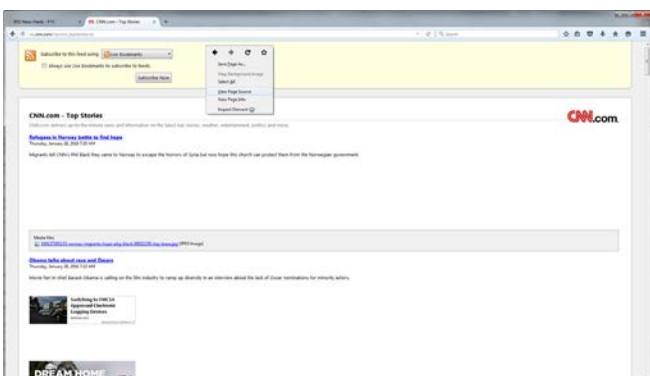
We've also included several additional tricks from our instructors that enable you to build services and applications more efficiently and effectively. Whether you're a rookie or a long-time user with vast experience using ThingWorx, you're likely unaware of these tricks. By taking advantage of these tricks, you'll be able to streamline your development efforts and create applications that build upon existing services while providing high levels of security.

## Tip 1: Leveraging ThingWorx Connectivity

This example illustrates how you can take advantage of ThingWorx connectivity to easily consume data from existing web services within ThingWorx applications. You'll create an application that grabs data hosted in a web service and use ThingWorx to display that data in a more graphical format.

Enter the URL [http://rss.cnn.com/cnn\\_topstories.rss](http://rss.cnn.com/cnn_topstories.rss), which takes you to Screen 1 below. If you open in Chrome, you will see the raw XML content. If you open in Firefox, right click anywhere on the page and select "View Page Source" to see the raw XML. Because the screen lacks an attractive user interface, you want to bring the data into ThingWorx to create a more user friendly presentation.

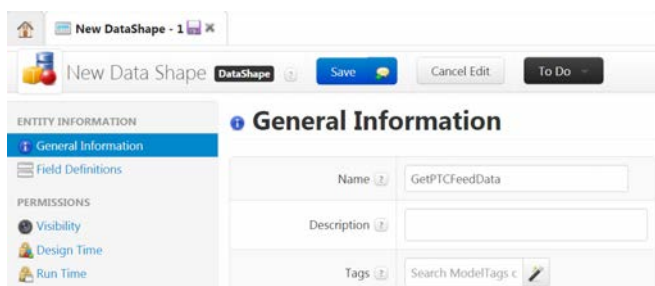
Screen 1



### Step 1: Create a Data Shape

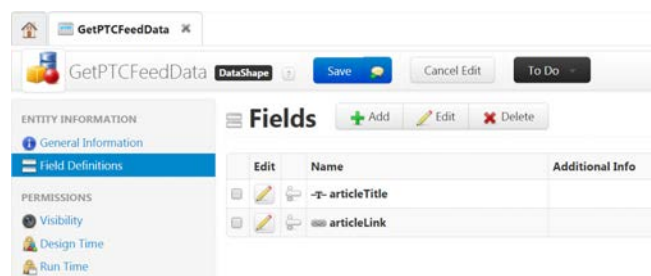
Start by creating a method that will output a table structure. Any time you need a table in ThingWorx, you must create a Data Shape to provide structure for the output. A Data Shape is an entity comprised of one or more fields that describe how the data is structured. Name the new Data Shape "GetPTCFeeData," as shown in Screen 2.

Screen 2



In ThingWorx's main Explorer window, go to the Entity Information Bar and select Field Definitions. When the Fields screen pops up, (Screen 3), define two fields for the Data Shape: articleTitle (String) and articleLink (Hyperlink) This step determines how the output will be put into table format. In this case the table will hold two fields, the title of the article and a link to it.

Screen 3

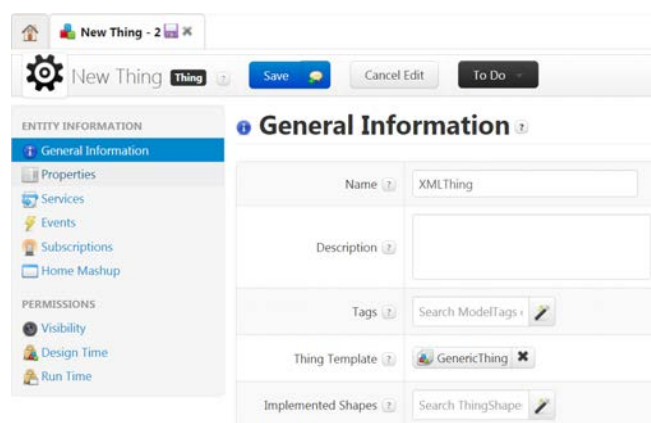


### Step 2: Create a New Service on a Thing

Now create a service that takes XML and outputs it. When you create a new service in ThingWorx, you do so on top of a Thing, which is the type of object that serves as ThingWorx's base entity. Therefore, you start by creating a new Thing.

In the Explorer menu, select New Thing, then click on General Information. Use this menu to create XMLThing, shown below in Screen 4.

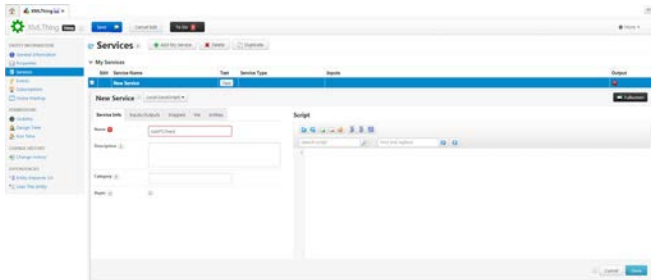
Screen 4



Now create a New Service on the Thing – called GetPTCFeeData. This service will load XML content, parse through it and output it as a table.

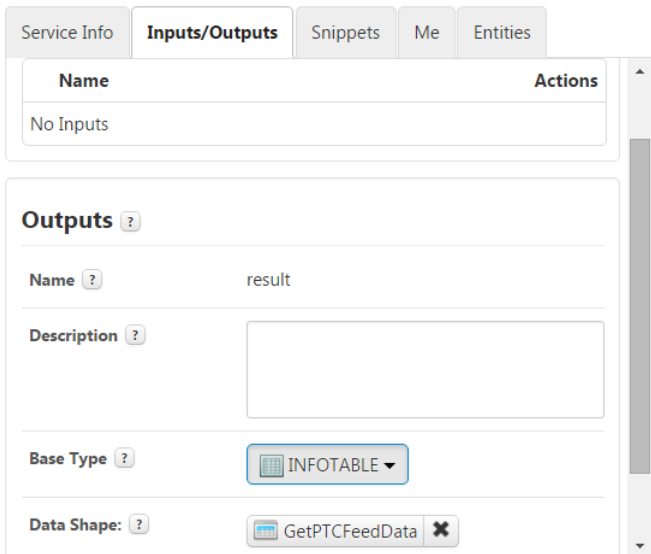
In the New Thing screen, go to the Entity Information bar and select “Services.” Call this new service GetPTCFeed, as seen below in Screen 5.

Screen 5



The service needs to know what the table should look like. We prepare the output of the service to be a table with two fields. In the New Service Page window (Screen 6), click on the Inputs/Outputs tab. Set the Output Base Type to INFOTABLE and then select the GetPTCFeedData, which is the Data Shape you created in Step 1.

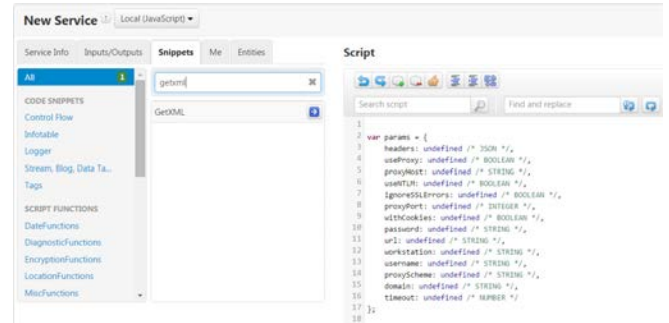
Screen 6



Now create a function to pull in a URL. Rather than requiring you to program this code from scratch, ThingWorx gives you the Snippets tool that generates the code for you.

Click on the Snippets tab to specify which Snippet you want to use, in this case “GetXML” (Screen 7), from a list of pre-defined Snippets.

Screen 7



In the code displayed in the window, replace “undefined” in the line that starts with “url...” with the following url: [“http://rss.cnn.com/rss/cnn\\_topstories.rss”](http://rss.cnn.com/rss/cnn_topstories.rss). This allows you to pass in the URL of the web service you want to consume within ThingWorx (Screen 8).

Screen 8

```

1
2 var params = {
3   headers: undefined /* JSON */,
4   useProxy: undefined /* BOOLEAN */,
5   proxyHost: undefined /* STRING */,
6   useNTLM: undefined /* BOOLEAN */,
7   ignoreSSLErrors: undefined /* BOOLEAN */,
8   proxyPort: undefined /* INTEGER */,
9   withCookies: undefined /* BOOLEAN */,
10  password: undefined /* STRING */,
11  url: "http://rss.cnn.com/rss/cnn_topstories.rss" /* STRING */,
12  workstation: undefined /* STRING */,
13  username: undefined /* STRING */,
14  proxyScheme: undefined /* STRING */,
15  domain: undefined /* STRING */,
16  timeout: undefined /* NUMBER */
17 };
18

```

As the default, this Snippet assumes you want the output of the service to be XML content. But instead, you want the output to be a table. Change the name of the last line of the Snippet from “var result = ...” to “var xmlPage = ...” (Screen 9) You’ll define your output later.

Screen 9

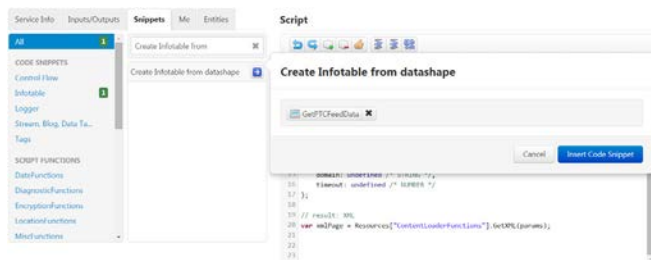
```

19 // result: XML
20 var xmlPage = Resources["ContentLoaderFunctions"].GetXML(params);
21

```

Now you start building the table. Select the Data Shape you created in Step 1. After the line referenced in Screen 9, bring in the Snippet called “Create Infotable from datashape” and select GetPTCFeeData (Screen 10). Since you just created this Data Shape, you can find it at the top of the list. This brings in code that uses the Data Shape to set the output to an empty table with two columns.

Screen 10



In Screen 11, add a “for loop” at the end of the code. The XML content you’ll be bringing in contains a number of objects/items that contain the name of the item and where it’s located. This “for” loop allows the service to find every place in the XML source page that talks about an item so it can look at each item.

Screen 11

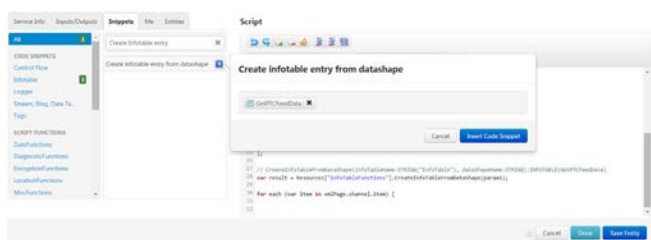
```

29
30 for each (var item in xmlPage.channel.item) {
31

```

The output table you just created is empty. You want to add a row to it. Start creating the structure to add a row. Inside the “for loop,” bring in the Snippet called “Create infotable entry from datashape.” (Screen 12). When you doubleclick on the name of that Snippet, the system asks you to choose a Data Shape. Select GetPTCFeeData

Screen 12



Now define what goes into the row. We replace “undefined” with “item.link” and “item.title” and type in “result” in the line below (Screen 13).

Screen 13

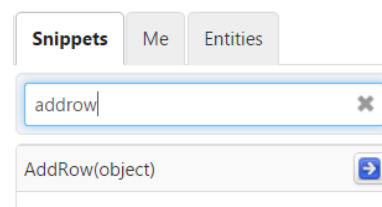
```

32 // GetPTCFeeData entry object
33 var newEntry = new Object();
34 newEntry.articleTitle = item.title; // STRING
35 newEntry.articleLink = item.link; // HYPERLINK
36 result
37

```

Add the row to the table. Screen 14 and 15 allow you to add a row to the output. In Screen 14, bring in the Snippet called “AddRow(object)” .

Screen 14



In the snipped code that appears, replace “yourRowObjectHere” (Screen 15) with “newEntry” and end the “for loop” on the next line.

Screen 15

```

30 for each (var item in xmlPage.channel.item) {
31
32 // GetPTCFeeData entry object
33 var newEntry = new Object();
34 newEntry.articleTitle = item.title; // STRING
35 newEntry.articleLink = item.link; // HYPERLINK
36 result.AddRow(newEntry);
37 }
38

```

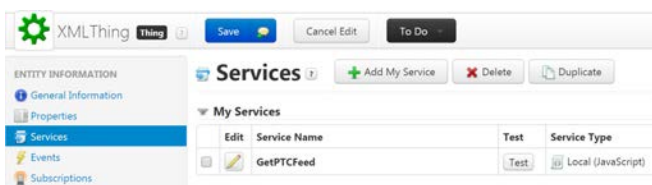
Click Done, then Save (Screens 16 and Screen 17).

Screen 16

```

34 newEntry.articleTitle = item.title; // STRING
35 newEntry.articleLink = item.link; // HYPERLINK
36 result.AddRow(newEntry);
37 }
    
```

Screen 17

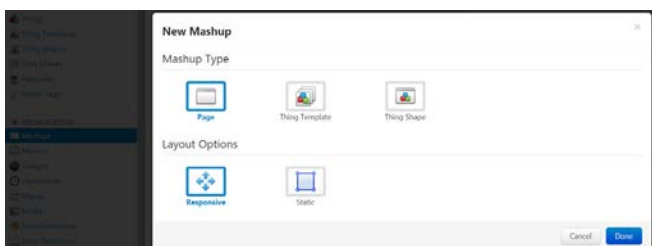


### Step 3: Create a New Mashup

Now create a Mashup to allow the user to see and interact with the output. You want this Mashup to show a list of articles on the left side of the screen and allow the user to click on one of these articles to display the actual web page on the right side of the screen.

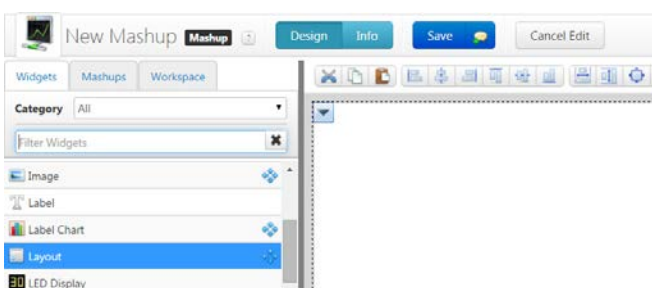
Create a new Mashup, specifying the Mashup Type as “Page” and the Layout Options as “Responsive.” Click Done (Screen 18).

Screen 18



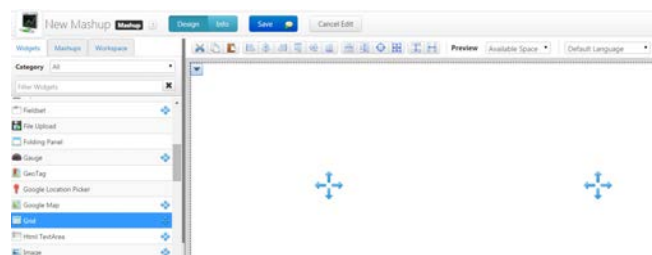
The New Mashup window appears. Drag and drop the Layout widget into the center workspace (Screen 19a), and click Done.

Screen 19a

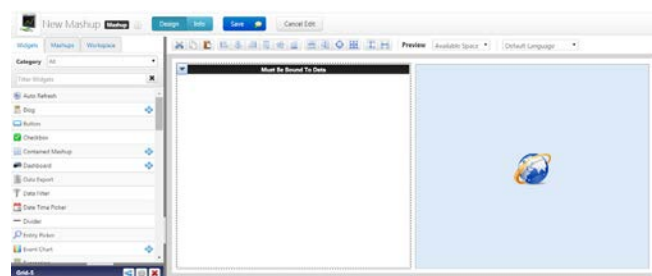


Drag and drop the Grid widget (Screen 20), which lets you display a table-- into the left side of the center workspace, and the Web Frame widget (Screen 21), which lets you open a web page in the context of a Mashup - into the right side.

Screen 20

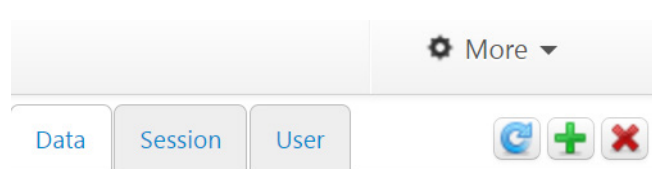


Screen 21



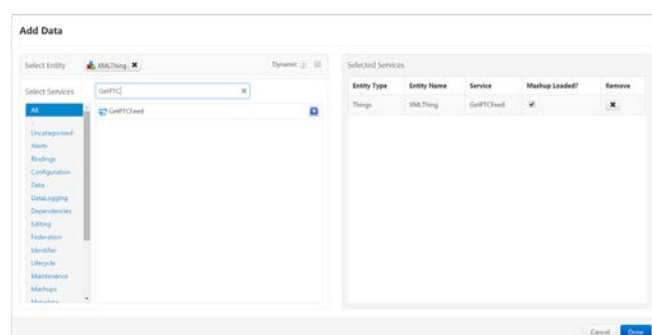
Click on the green plus in the top right to add the entity (Screen 22).

Screen 22



Select the service you created previously to load the XML content. In Screen 23, select the XMLThing entity, then use the filter to find GetPTCFeed, check “Mashup Loaded?”, and hit Done.

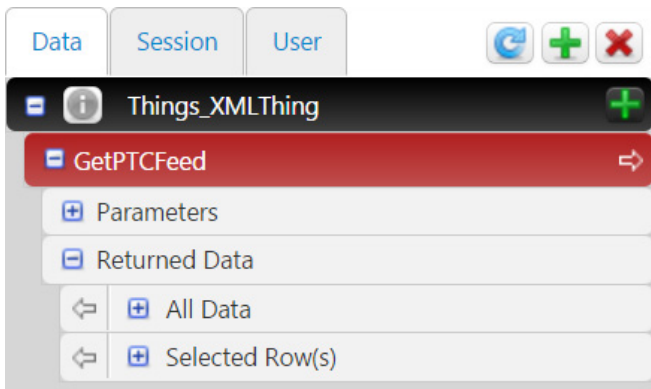
Screen 23



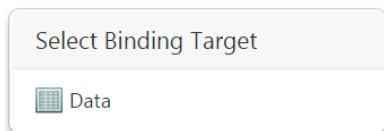


Now you want to show the output of the service in the grid. From the right side of the Mashup screen (Screen 24a), drag and drop All Data onto the Grid and select “Data” (Screen 24b)

Screen 24a

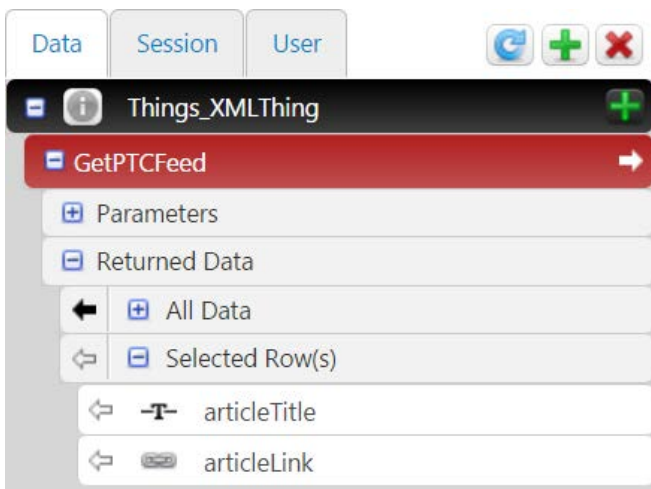


Screen 24b



When the user selects an article in the grid, you want him to see that article in the Web frame. From the right side of the Mashup Window, expand Selected Row(s) - articleLink and articleTitle appear (Screen 25a). Drag and drop articleLink onto the Web Frame and select “URL” as the binding target (Screen 25b).

Screen 25a



Screen 25b



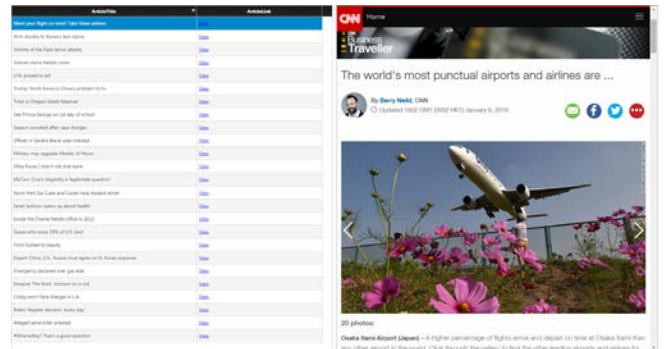
Click Save, name the Mashup – PTCFeedMashup – and click Save again in Screen 26.

Screen 26



To see the end result, click View Mashup and select an article. You see it pop up in the Mashup screen (Screen 27).

Screen 27



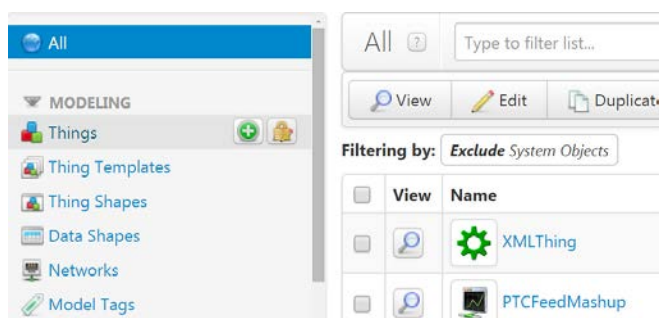
## Tip 2: Using Tags to Group and More Easily Find Related Items

Many developers work on multiple projects at once. If you're developing, say, five projects simultaneously, and you're ready to put one of these projects into production, it can be time consuming to determine exactly what items comprise any one solution.

With ThingWorx, you can use Tags to easily keep track of all the items used in each project. Once an item has been tagged, you can use a filter to find everything with the same tag.

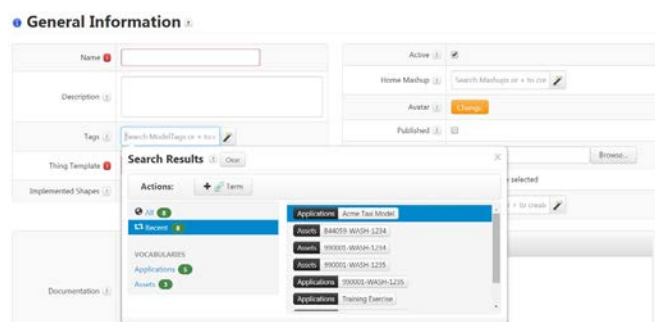
To create a Tag, start at the Explorer Menu, shown in Screen 28, go to the Modeling menu and select Things. Click on the green plus sign that appears.

Screen 28



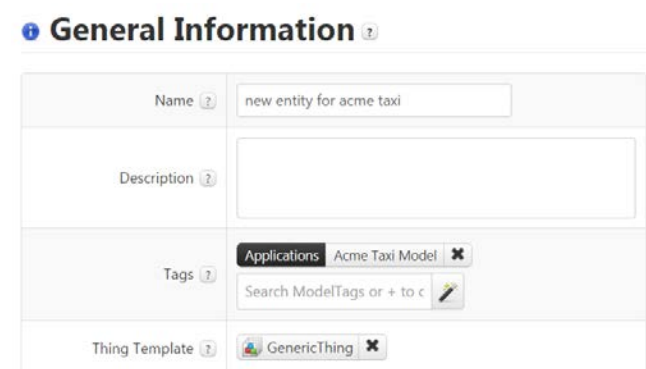
When the General Information menu, Screen 29, pops up, select Edit from the Tags menu. The Search Results window pops up, showing a list of existing tags. In this example, choose the "Acme Taxi Model" Tag.

Screen 29



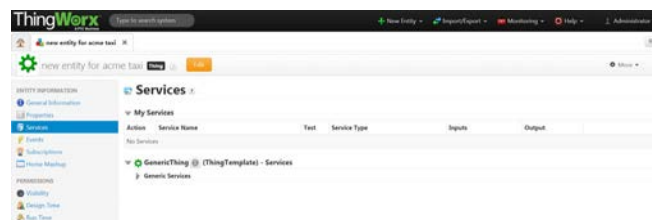
Now assign the AcmeTaxiModel tag to a new entity. To create that entity, go to Name in the General Information screen (Screen 30) and type in "new entity for Acme Taxi." Whenever you create a new entity in ThingWorx, it must be based on a Thing template. For this example, use the GenericThing template by going to the Thing Template, selecting Generic Thing, and Save.

Screen 30



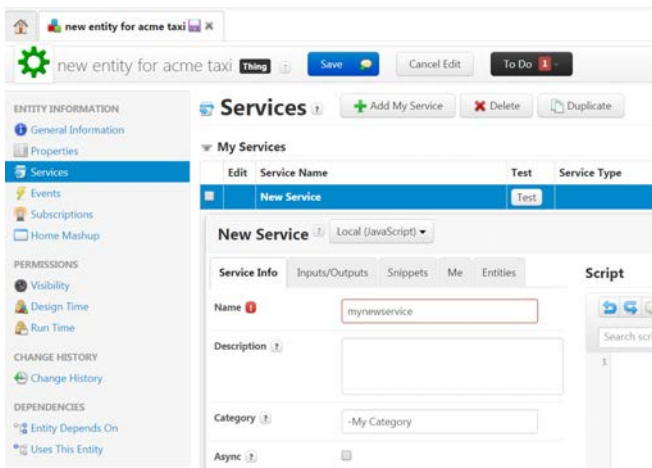
Now you want to group your services to sort them more easily. Start by creating a new Service. Go to the Entity Information menu, select Services and Edit. Screen 31 will appear.

Screen 31



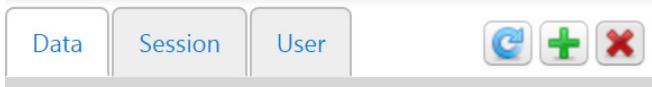
In the New Services screen that pops up, type in the name, in this case "mynewservice" (Screen 32). In the Category window type in MyCategory, preceded by a dash, click Done and Save. The service mynewservice is now associated with the tag -My Category.

Screen 32



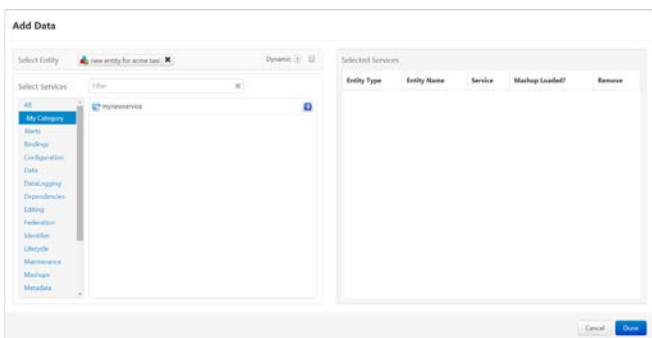
To find the newly created service, go to the top right hand corner of the Mashup Builder. When you select Data, you'll see a list of of services, select TW.RSM. Mashup.wiki. and press the green plus sign, Screen 33.

Screen 33



The Add Data screen (Screen 34) pops up. You can easily find your newly created Thing, new entity for Acme Taxi, because the tag “ -My Category” appears at the top of the list of services. This occurs because ThingWorx sorts the list alphabetically.

Screen 34

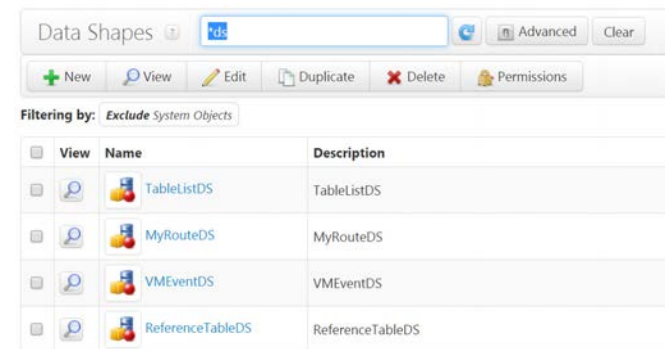


### Tip 3: Set Up Naming Conventions

When a team is developing in ThingWorx, a best practice is to use and enforce naming conventions to take the mystery out of other people’s logic, create order and streamline workflows. Yet, basic training often doesn’t highlight the need to set up naming conventions. Customers who are not in big shops may not have established naming standards.

As you develop a logical naming convention, use a logical appendix that indicates the nature of the object. For example, you might include “Shape” in the name if the entity is a shape. Consider using short hand, such as the abbreviation DS for DataSheet (Screen 35). And don’t forget that ThingWorx names can contain periods.

Screen 35



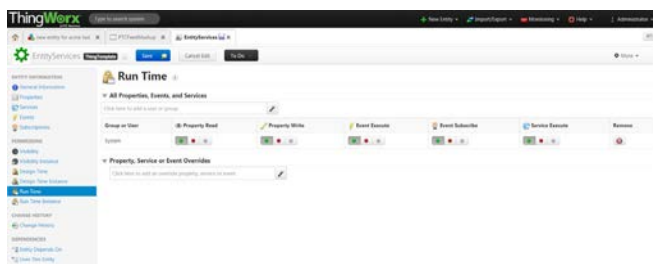


### Tip 4: Provide Pass-Through Permissions to Limit Use of a Service

ThingWorx contains many useful reusable services. But giving users access to all of your services' functionality is a security risk. To mitigate this risk, ThingWorx provides the System User, which provides pass-through permissions to run a service. By taking advantage of the System User, you can enable end users to indirectly access the service to take advantage of service functions on a need-to-know basis or to execute the service in a limited fashion. For example, if there's a service called "Get all the things", you can use the System User to leverage this service to allow different users to only "get my things."

To enable this capability, start by assigning permissions to the System User. Go to the ThingTemplate for your Entities. Go to Runtime Instance to set security on all Things that implement the ThingTemplate. In ALL Properties, Services and Events in Screen 36, select the System User and give that user a full set of permissions— Property Read, Property Write, Service, Event and Subscription Execute. Now the System user can give permission to all services for the Things that implement the ThingTemplate.

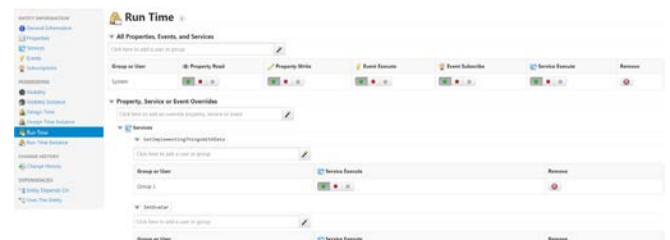
Screen 36



Next, you'll allow a user to leverage the service without allowing them to use the services freely. Create a service that uses some, but not all of the capabilities in the original service.

Finally, grant the user appropriate permissions. Go to the Override section, find the Service that the User will use in their Mashup, select it, and then select the user or group. Give that group or user permission to execute the Service (Screen 37).

Screen 37



### Tip 5: Learn How an Existing Thing was Created

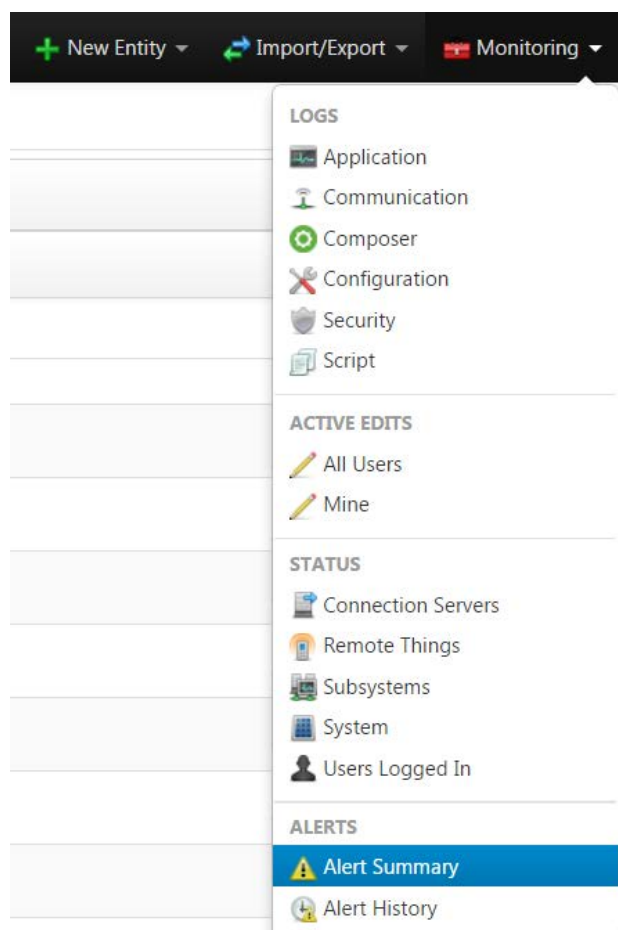
A great way to learn how to develop new Services or Mashups in ThingWorx is to look at how existing Things were created and then modify them for your own purposes.

For example, ThingWorx comes with a service that provides a monitoring screen with an alert summary that allows users to see and manage all their alerts. Let's assume you want to create a similar alert screen in a Mashup with several modifications to meet your own requirements.

This tip shows you how you can use the Duplication capability to see how the existing Alert screen was created. You can then modify the existing service rather than having to create a new one from scratch.

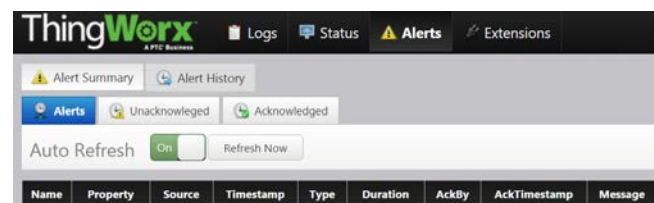
Start by going to the MyMashups menu and Select the Alerts Summary as shown in Screen 38.

Screen 38



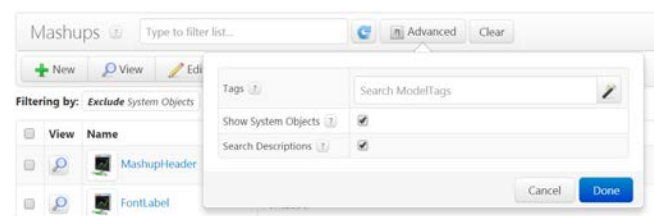
You need to know the name of the service. You can find the name in the URL. When Screen 39 pops up, select Auto Refresh to see the URL.

Screen 39



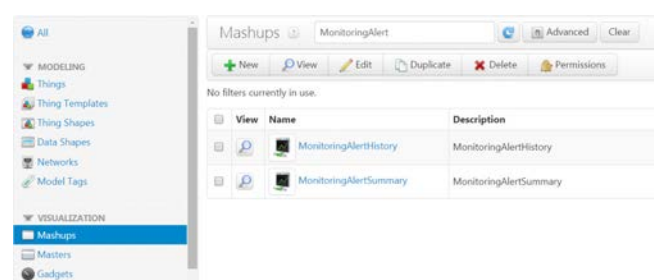
Since this is a System Object, go to the Explorer menu, select Mashups in the Visualization menu and check the box to Show System Objects (Screen 40).

Screen 40



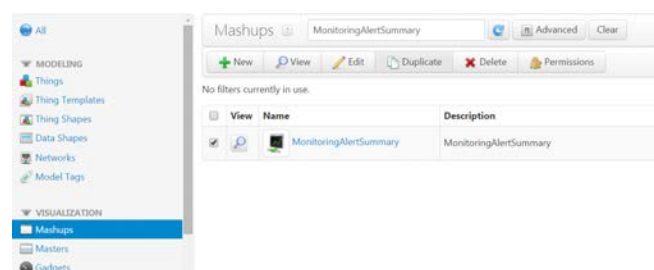
In Screen 41, input in the name of the Mashup. In this case, copy the name you found in the URL in Screen 39, "MonitoringAlertSummary." When the Filters menu pops up, click on the filter.

Screen 41



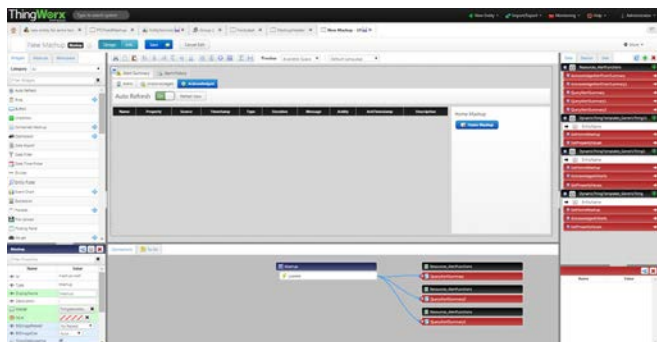
Click on Duplicate, as shown in Screen 42.

Screen 42



Screen 43 appears. The items in the red and black boxes show the services ThingWorx used to create the Alerts Mashup and the front-end logic that relates them all together. You can modify or mix-and-match these services and logic in different ways to build a new object or entity.

Screen 43



## Conclusion

These tips and tricks show you how to take advantage of ThingWorx’s powerful functionality and provide ways to improve productivity and security. These are all representative of the types of productivity improvements that interacting with instructors during PTC University training sessions will bring to light.

PTC University’s training and tools help individuals and organizations use PTC software more effectively. Whether you’re looking for specific training courses, ways to improve the proficiency of experienced users, or a personalized corporate learning program that meet your training schedule and budget requirements, PTC University can help you optimize your business through traditional classroom training, online course, as well as leadership training and consulting. PTC University enables you to make the most of your software investment through:

**PTC Product Knowledge:** First Hand. PTC University has direct access to the PTC product development and consulting organizations to ensure that our courses include the most up-to-date product information plus best practices derived from thousands of customer deployments.

**Global Expert Network:** All PTC University training classes are delivered by experienced instructors or certified training partners. Our global network is delivered with consistent quality no matter where you’re located.


**Flexible Delivery Options:** PTC University offers any instructor-led class in the delivery format of your choice, on-site, virtual, or in a traditional classroom, and all ThingWorx classes are available as eLearning libraries as well.

To learn more about how our training can help your organization, please [contact us today](#). Our Training Advisors will be happy to help you select the right training option for you.

Visit the PTC University website:  
[PTC.com/services/training](http://PTC.com/services/training)

© 2016, PTC Inc. All rights reserved. Information described herein is furnished for informational use only, is subject to change without notice, and should not be taken as a guarantee, commitment, condition or offer by PTC. PTC, the PTC logo, Product & Service Advantage, Creo, Elements/Direct, Windchill, Mathcad, Arbortext, PTC Integrity, Servigistics, ThingWorx, ProductCloud and all other PTC product names and logos are trademarks or registered trademarks of PTC and/or its subsidiaries in the United States and other countries. All other product or company names are property of their respective owners.

J06417-ThingWorx Tips and Tricks-EN-0116



NET AG system integration  
 Schellerdamm 16  
 21079 Hamburg  
 www.net-online.de

Phone: +49 (0)800 NET4YOU  
 Phone: +49 (0)800 638 48 68  
 Fax: +49 (0)40 767 302 63  
 kontakt@net-online.de