Working with and Developing with Managed Metadata

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\14_ECM\Lab

Lab Overview: In this module you will gain hands-on experience programming against the SharePoint 2016 Managed Metadata service application taxonomies. The previous version of SharePoint only provided a server side API for developing against taxonomies in the Managed Metadata service application. The SharePoint 2016 includes a CSOM interface for working with taxonomies.

Exercise 1: Creating & Using Taxonomies with the Managed Metadata Term Store Tool

In this exercise you will create a new taxonomy using the Managed Metadata term store tool and use it within a site.

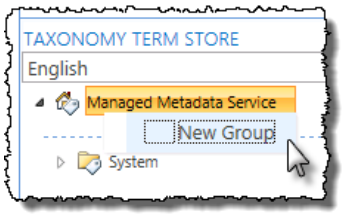
1. Setup a new site collection for this lab:
   a) Ensure you are logged into the **WingtipServer** server as **WINGTIP\Administrator**.
   b) Run a PowerShell script, found in the root lab folder for this module:
      i) Right-click **SetupModule.ps1** and select **Run with PowerShell**. This file can be found in the files associated with this lab:

      ```
      C:\Student\Modules\14_ECM\Lab
      ```

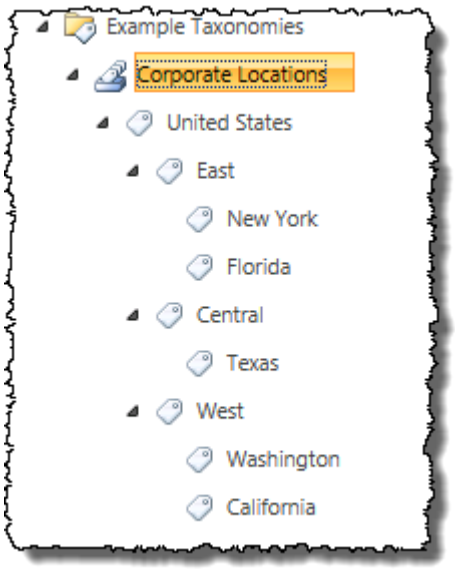
   c) When the script completes, it will launch a new browser and navigate to the lab site collection: https://ecm.wingtip.com.
   d) Close the PowerShell console window.
2. Connect and configure the Managed Metadata Service for the remainder of this lab:
   a) Open **Central Administration**:
   b) Go to the **Term Store Management Tool** page:
      i) Application Management → Manage Service Applications → Managed Metadata Service
   c) When the **Term Store Management Tool** page loads, enter **WINGTIP\Administrator** in the **Term Store Administrators** box and click the person with a check mark icon **"Check Names"** just below to the box to resolve the user.
   d) Click **Save** at the bottom of the page.

Create a New Taxonomy using the Term Store Tool

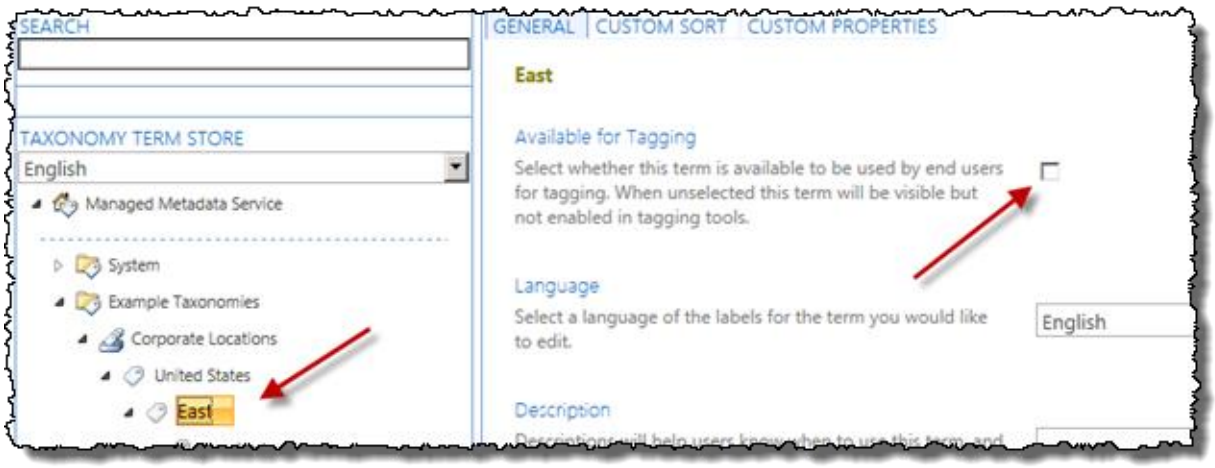
3. Within the **Term Store Management Tool**, select the **Managed Metadata Service** node in the left-hand part of the page and select **New Group**.



4. Set the name of the group to **Example Taxonomies**.
5. Select the **Example Taxonomies** group and select the drop-down **New Term Set**.
6. Set the name of the term set to **Corporate Locations**.
7. Select **Corporate Locations** term set and select **Create Term**. Use this technique to create a small taxonomy as shown in the following figure:



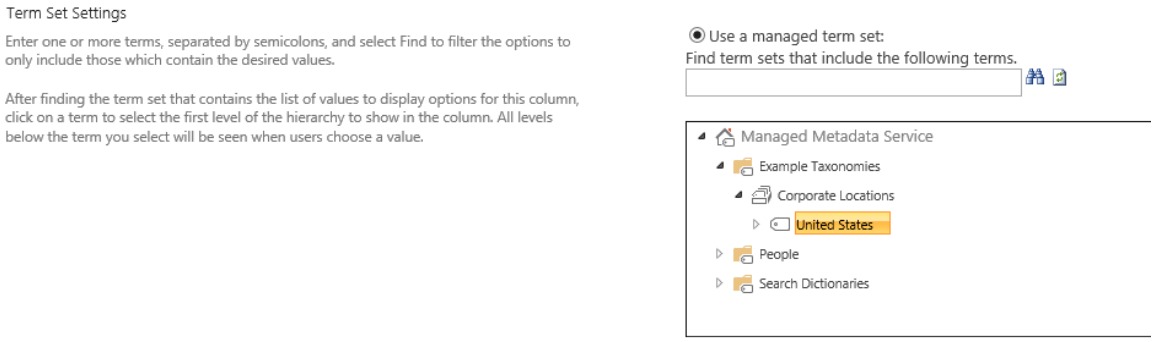
8. Now, select the **East** term. This term should not be available for tagging as it is only used to group terms. In the right-hand pane you will find the detail for this term.
9. Uncheck the **Available for Tagging** checkbox and click **Save**.



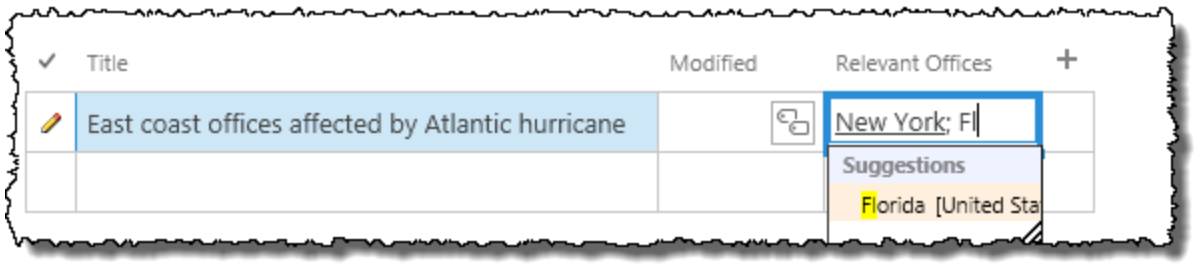
10. A new capability of the Term Store Management Tool is the ability to visually edit the custom properties on a term. Add a new property to signify the primary point of contact for a corporate region:
    a) Select the **Texas** term.
    b) In the right-hand pane, select the **Custom Properties** tab at the top.
    c) Under the **Shared Properties**, select **Add**.
    d) Name the property **PrimaryPOC** and give it a value of **Ken Sanchez** and click **Save**.

Use the Taxonomy in a SharePoint List

11. Next, utilize the taxonomy you just created in a SharePoint site. Using Internet Explorer, navigate to http://ecm.wingtip.com.
12. Create a new Announcements list:
    a) In the **Quick Launch** navigation on the left-hand side of the page, click **Site Contents**.
    b) On the **Site Contents** page, under the **Lists, Libraries and Other Apps** section, click the **Add an App** link.
    c) On the **Site Contents > Your Apps** page, select **Announcements**.
    d) Set the name of the list to **Announcements**.
13. Modify the **Announcements** list to contain terms for the different company locations:
    a) In the **Quick Launch** navigation, select the **Announcements** list/app.
    b) When the **Announcements** list loads, using the ribbon select the **List** tab and click the **List Settings** button.
    c) Click the **Create column** link in the **Columns** section of the page.
    d) On the **Settings > Create Column** page, use the following to create a new column mapped to the taxonomy you created and click **OK**:
        i) **Column Name:** Relevant Offices
        ii) **The type of information in this column is:** Managed Metadata
        iii) **Allow multiple values**: checked
        iv) **Term Set Settings**: Use a managed term set
            (1) Select the **United States** term set as shown in the following figure:



14. Now add some data to your list:
    a) In the **Quick Launch** navigation, select the **Announcements** list.
    b) Add an announcement to the list.
    c) Next, add multiple announcements, except this time use the **Edit** view. Using the ribbon, click the **List** tab and the **Quick Edit** button in the **View Format** group. This was previously called the Datasheet View. It has been improved to support managed metadata!
    d) Add five – ten announcements using the **Quick Edit** view and make sure each has a term tagged to it. Some should have multiple offices tagged.



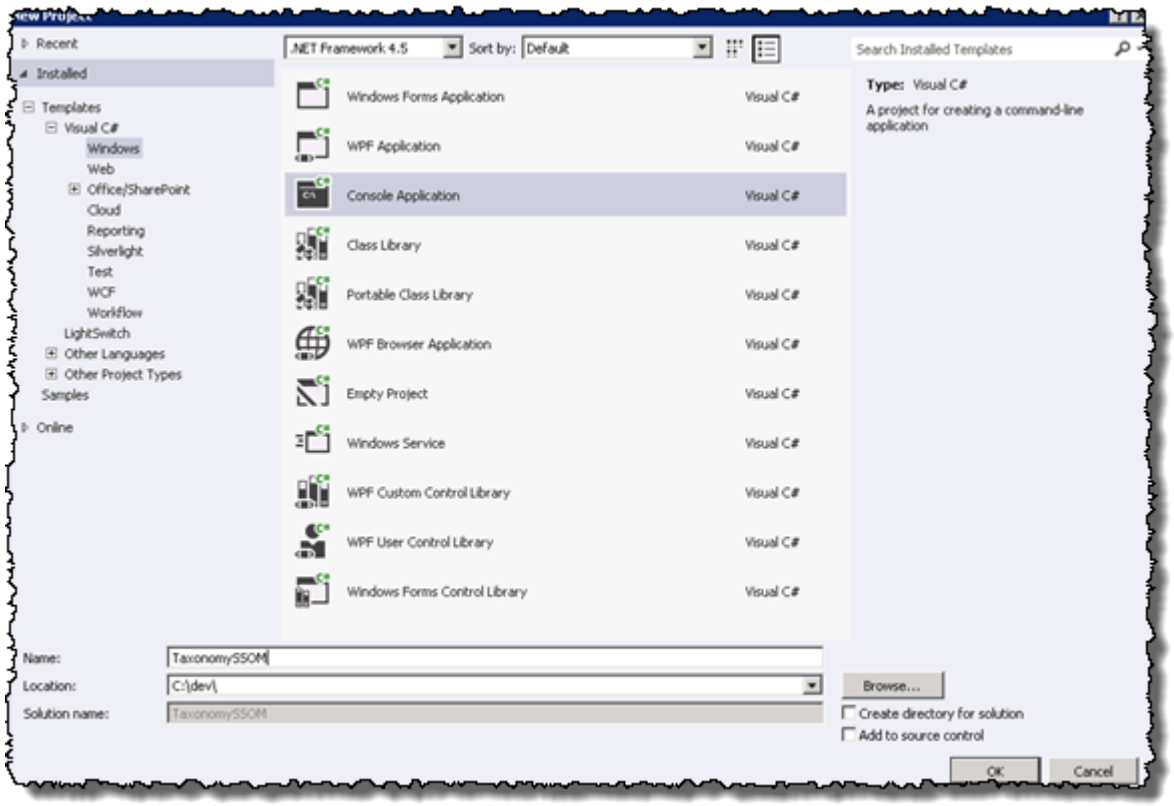
15. Last, see how the managed navigation works. First, enable it on site:
    a) Using the **Site Actions** "gear" icon in the top-right corner of the browser, select **Site Settings**.
    b) In the **Site Actions** setting, select **Manage Site Features**.
    c) Find the Site Feature **Metadata Navigation and Filtering** and click the **Activate** button.
16. Now enable it on the list:
    a) In the **Quick Launch** navigation, select the **Announcements** list/app.
    b) When the **Announcements** list loads, using the ribbon select the **List** tab and click the **List Settings** button.
    c) Under the **General Settings** section, select **Metadata navigation settings**.
    d) In the **Configure Navigation Hierarchies**, add the **Relevant Offices** column from the **Available Hierarchy Fields** to the **Selected Hierarchy Fields** and click **OK**.
17. Next, go back to the **Announcements** list.
18. Notice in the lower portion of the **Quick Launch** you have a new navigation option. Using this you can filter all the content in the list, regardless of any folders you may have.

In this exercise you created a term set using the browser interface and used it within a SharePoint list.

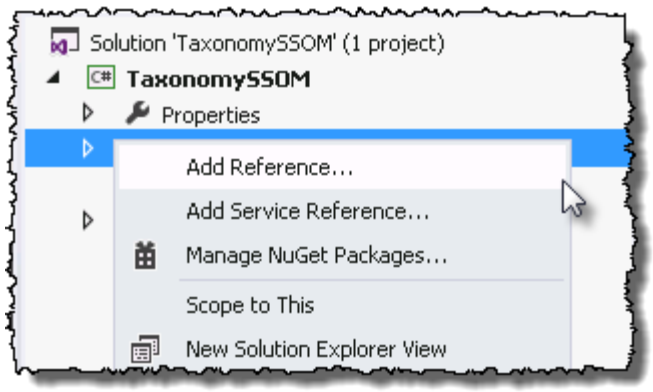
Exercise 2: Query and Update the Taxonomy Using the Server-Side API

In this exercise you will create a console application that will query and update the taxonomy using the SharePoint server-side object model.

1. Launch **Visual Studio** by selecting **Start → All Programs → Microsoft Visual Studio 2012 → Visual Studio 2012**.
2. Create a new project:
   a) Select **File → New → Project**.
   b) In the left-hand section of the **New Project** dialog, select **Templates → Visual C# → Windows → Console Application**.
   c) In the top part of the dialog, select **.NET Framework 4.5**.
   d) Give the project a name of **TaxonomySSOM**.



   e) Click **OK** to create the project.
3. In order to work with the taxonomy API you need to add a few assembly references to the project:
   a) Within the **Solution Explorer** tool window, right-click **References** in the project and select **Add Reference…**.



   b) Click the **Browse** button at the bottom of the dialog and add the two assemblies found in the following folder.

```
c:\Program Files\Common Files\Web Server Extentions\15\ISAPI
```

i) **Microsoft.SharePoint.dll**

ii) **Microsoft.SharePoint.Taxonomy.dll**

c) Click **OK** at the bottom of the dialog.

4. Finally, ensure this console application is a 64-bit application as SharePoint is 64-bit:
   a) Right-click the project in the **Solution Explorer** and select **Properties**.
   b) On the **Build** tab, select **Platform Target = x64**.
5. Save the project by clicking **File → Save All**.

Create Terms in the Term Set

6. Within the **Program.cs** file, add two using statements to save typing out the full type name later in the app:

```
using Microsoft.SharePoint;
using Microsoft.SharePoint.Taxonomy;
```

7. Now go into the **Main()** method. You need to get a reference to the term set previously created so the first step is to connect to a site collection, establish a taxonomy session and connect to the term set. Do this by adding the following code just inside the **Main()** method:

```
SPSite siteCollection = new SPSite("http://ecm.wingtip.com");
TaxonomySession taxSession = new TaxonomySession(siteCollection);
TermStore termStore = taxSession.TermStores[0];
Group termGroup = termStore.Groups.Single(tg => tg.Name == "Example Taxonomies");
TermSet termSet = termGroup.TermSets.Single(ts => ts.Name == "Corporate Locations");
```

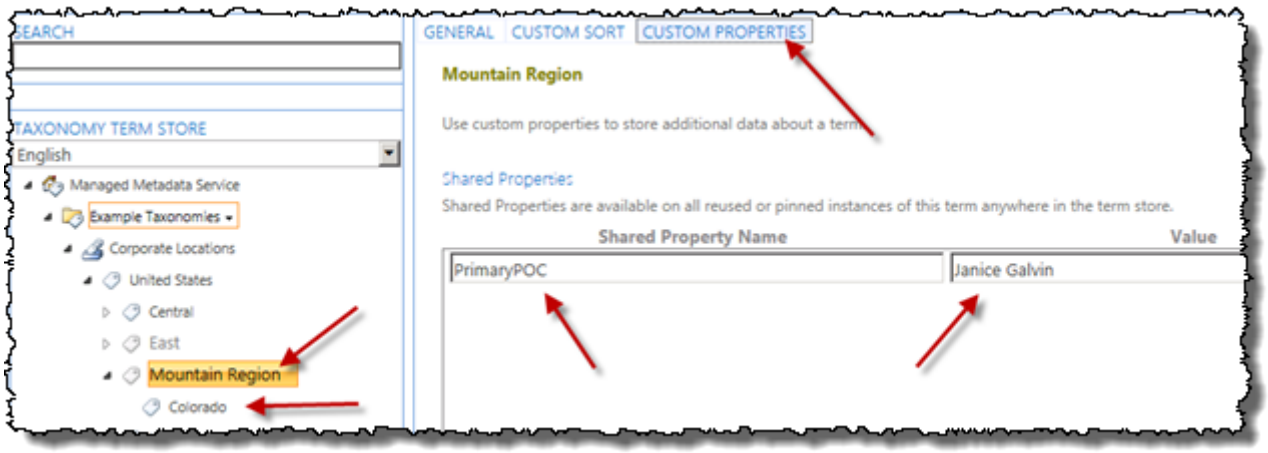
8. With a reference to the term set, create a new region term and add a location to it. Add the following code after the code you just added to the **Main()** method:

```
// add region
Term unitedStatesTerm = termSet.Terms[0];
Term newRegion = unitedStatesTerm.CreateTerm("Mountain Region", 1033);
newRegion.SetCustomProperty("PrimaryPOC", "Janice Galvin");
newRegion.IsAvailableForTagging = false;

// add state
Term newState = newRegion.CreateTerm("Colorado", 1033);

// save changes
termStore.CommitAll();
```

9. Save the project by clicking **File → Save All**.

Deploy and Test the Project

10. Deploy and test your application by pressing **[F5]** or **Debug → Start Debugging**. This will compile and execute the app. You will see the console application launch with a black box but because it doesn't write anything to the console, nothing will be seen.
11. When the console app completes and the black box closes, check the changes you made using the Term Store Management Tool:
    a) Open **Central Administration**:
       i) **Start → All Programs → Microsoft SharePoint 2013 Products → SharePoint 2013 Central Administration**
    b) Go to the **Term Store Management Tool** page:
       i) **Application Management → Manage Service Applications → Managed Metadata Service**
12. Expand the **Example Taxonomies** group and find the terms you just added:



Modify the Project to Update Terms

13. The previous task had you run and test the console app. If you did that again you would get an error as it would try to recreate terms with the same name.
    a) Comment out all the code that added a new region & state from the app. The **Program.cs** file's **Main()** method should look like this:

```
SPSite siteCollection = new SPSite("http://ecm.wingtip.com");
TaxonomySession taxSession = new TaxonomySession(siteCollection);
TermStore termStore = taxSession.TermStores[0];
Group termGroup = termStore.Groups.Single(tg => tg.Name == "Example Taxonomies");
TermSet termSet = termGroup.TermSets.Single(ts => ts.Name == "Corporate Locations");

/* START COMMENT
// add region
Term unitedStatesTerm = termSet.Terms[0];
Term newRegion = unitedStatesTerm.CreateTerm("Mountain Region", 1033);
newRegion.SetCustomProperty("PrimaryPOC", "Janice Galvin");
newRegion.IsAvailableForTagging = false;

// add state
Term newState = newRegion.CreateTerm("Colorado", 1033);
   STOP COMMENT */

// save changes
termStore.CommitAll();
```

14. With the term creation code commented out, add some code to find a term. Insert the following code before the ***// save changes*** comment in the **Main()** method:

```
// search for CENTRAL term
var foundTerm = termSet.GetTerms("Central", 1033, false).First();
```

15. After finding the term, add code to update the title to include "Region" and to disallow tagging for the term:

```
foundTerm.Name = "Central Region";
foundTerm.IsAvailableForTagging = false;
```

16. Repeat the same process for the other two regions that were manually added:

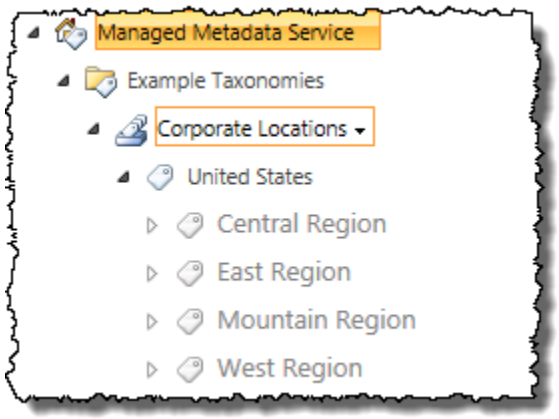
```
// search for EAST term
foundTerm = termSet.GetTerms("East", 1033, false).First();
foundTerm.Name = "East Region";
foundTerm.IsAvailableForTagging = false;
```

```
// search for WEST term
foundTerm = termSet.GetTerms("West", 1033, false).First();
foundTerm.Name = "West Region";
foundTerm.IsAvailableForTagging = false;
```

17. Save the project by clicking **File → Save All**.

Deploy and Test the Project

18. Test your changes by pressing **[F5]** or **Debug → Start Debugging**. This will compile and execute the app. You will see the console application launch with a black box but because it doesn't write anything to the console, nothing will be seen.
19. When the console app completes and the black box closes, check the changes you just made using the Term Store Management Tool:
    a) Open **Central Administration**:
       i) **Start → All Programs → Microsoft SharePoint 2013 Products → SharePoint 2013 Central Administration**
    b) Go to the **Term Store Management Tool** page:
       i) **Application Management → Manage Service Applications → Managed Metadata Service**
    c) In the list of service applications, select **Managed Metadata Service** to launch the **Term Store Management Tool**.
    d) Expand the **Example Taxonomies** group and find the terms you just modified:



In this exercise you worked with the SharePoint server-side object model for taxonomies.

Exercise 3: Query and Update the Taxonomy Using the Managed CSOM

In this exercise you will create a console application that will query and update the taxonomy using the SharePoint managed client side object model (CSOM).

20. If Visual Studio 2012 is not already open then Launch Visual Studio:
    a) **Start → All Programs → Microsoft Visual Studio 2012 → Visual Studio 2012**
21. Create a new project:
    a) Select **File → New → Project**.
    b) In the left-hand section of the **New Project** dialog, select **Templates → Visual C# → Windows → Console Application**.
    c) In the top part of the dialog, select **.NET Framework 4.5**.
    d) Give the project a name of **TaxonomyManagedCSOM**.
    e) Click **OK** to create the project.
22. In order to work with the taxonomy API you need to add a few assembly references to the project:
    a) Within the **Solution Explorer** tool window, right-click References in the project and select **Add Reference…**.

b) Click the **Browse** button at the bottom of the dialog and add the two assemblies found in the following folder:

```
c:\Program Files\Common Files\Web Server Extentions\15\ISAPI folder:
```

i) **Microsoft.SharePoint.Client.dll**

ii) **Microsoft.SharePoint.Client.Runtime.dll**

iii) **Microsoft.SharePoint.Client.Taxonomy.dll**

c) Click **OK** at the bottom of the dialog.

23. Save the project by clicking **File → Save All**.

Create Terms in the Term Set

24. Within the **Program.cs** file, add two using statements to save typing out the full type name later in the app:

```
using Microsoft.SharePoint.Client;
using Microsoft.SharePoint.Client.Taxonomy;
```

25. Now go into the **Main()** method. You need to get a reference to the term set previously created so the first step is to connect to a site collection, establish a taxonomy session and connect to the term set. This all must be done through the CSOM's ClientContext object. Do this by adding the following code just inside the **Main()** method:

```
ClientContext context = new ClientContext("http://ecm.wingtip.com");

TaxonomySession session = TaxonomySession.GetTaxonomySession(context);
context.Load(session, taxSession => taxSession.TermStores.Include(
         taxStore => taxStore.Groups.Include(
         taxGroup => taxGroup.TermSets.Include(tax => tax.Name)
         )));
context.ExecuteQuery();

TermStore termStore = session.TermStores[0];
TermGroup termGroup = termStore.Groups[0];
TermSet termSet = termGroup.TermSets[0];
```

26. With a reference to the term set, create a new region term and add a location to it. Add the following code after the code you just added to the **Main()** method:

```
// get UNITED STATES term
var terms = termSet.Terms;
context.Load(terms);
context.ExecuteQuery();
Term unitedStatesTerm = terms[0];
context.Load(unitedStatesTerm);
context.ExecuteQuery();

// add region
Term newRegion = unitedStatesTerm.CreateTerm("Pacific", 1033, Guid.NewGuid());
newRegion.SetCustomProperty("PrimaryPOC", "Rob Walters");
newRegion.IsAvailableForTagging = false;

// add state
Term newState = newRegion.CreateTerm("Hawaii", 1033, Guid.NewGuid());

// save changes
termStore.CommitAll();
context.ExecuteQuery();
```

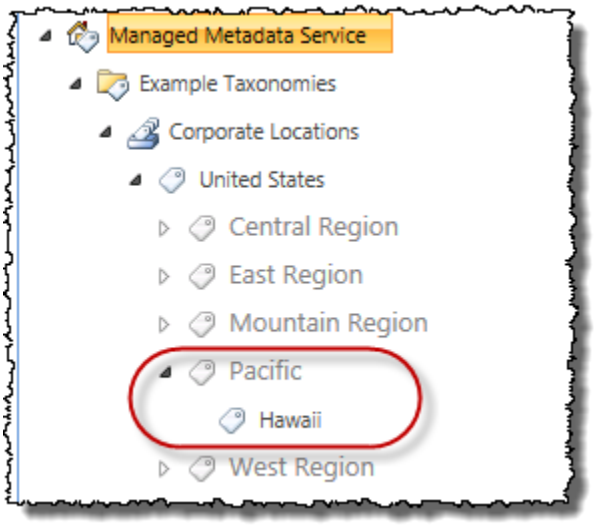
27. Save the project by clicking **File → Save All**.

Deploy and Test the Project

28. Test your changes by pressing **[F5]** or **Debug → Start Debugging**. This will compile and execute the app. You will see the console application launch with a black box but because it doesn't write anything to the console, nothing will be seen.

29. When the console app completes and the black box closes, check the changes you made using the Term Store Management Tool:

a) Open **Central Administration**:
   i) **Start → All Programs → Microsoft SharePoint 2013 Products → SharePoint 2013 Central Administration**
b) Go to the **Term Store Management Tool** page:
   i) **Application Management → Manage Service Applications → Managed Metadata Service**
c) In the list of service applications, select **Managed Metadata Service** to launch the **Term Store Management Tool**.
d) Expand **the Example Taxonomies** group and find the terms you just modified:



Modify the Project to Query for Terms

30. The previous task had you run and test the console app. If you did that again you would get an error as it would try to recreate terms with the same name, so comment out all the code that added a new region and state from the app. The **Program.cs** file's **Main()** method should look like this:

```
ClientContext context = new ClientContext("http://ecm.wingtip.com");

TaxonomySession session = TaxonomySession.GetTaxonomySession(context);
context.Load(session, taxSession => taxSession.TermStores.Include(
         taxStore => taxStore.Groups.Include(
         taxGroup => taxGroup.TermSets.Include(tax => tax.Name)
         )));
context.ExecuteQuery();

TermStore termStore = session.TermStores[0];
TermGroup termGroup = termStore.Groups[0];
TermSet termSet = termGroup.TermSets[0];

/* START COMMENT
// get UNITED STATES term
var terms = termSet.Terms;
context.Load(terms);
context.ExecuteQuery();
Term unitedStatesTerm = terms[0];
context.Load(unitedStatesTerm);
context.ExecuteQuery();

// add region
Term newRegion = unitedStatesTerm.CreateTerm("Pacific", 1033, Guid.NewGuid());
newRegion.SetCustomProperty("PrimaryPOC", "Rob Walters");
newRegion.IsAvailableForTagging = false;

// add state
Term newState = newRegion.CreateTerm("Hawaii", 1033, Guid.NewGuid());
   STOP COMMENT */
```

```
// save changes
termStore.CommitAll();
context.ExecuteQuery();
```

31. With the term creation code commented out, add some code to find a term. Insert the following code before the **// save changes** comment in the **Main()** method:

```
// search for PACIFIC term
var searchQuery = new LabelMatchInformation(context) {
   TermLabel = "Pacific",
   TrimUnavailable = false
};
var foundTerms = termSet.GetTerms(searchQuery);
context.Load(foundTerms);
context.ExecuteQuery();
```

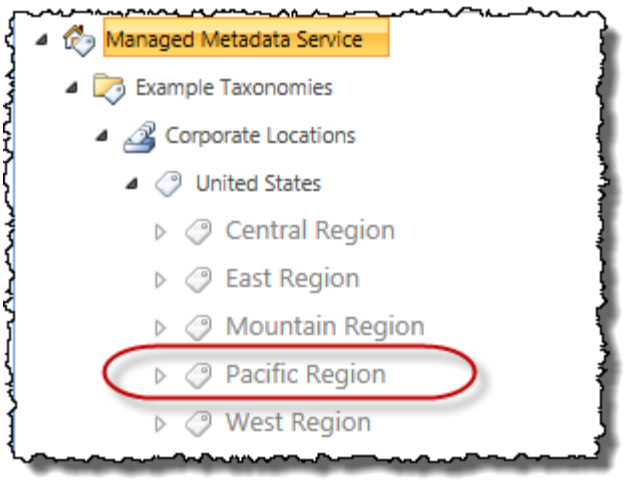
32. After finding the term, add code to update the title to include “Region” and to disallow tagging for the term:

```
// update term
foundTerms[0].Name = "Pacific Region";
```

33. Save the project by clicking **File → Save All**.

Deploy and Test the Project

34. Test your changes by pressing **[F5]** or **Debug → Start Debugging**. This will compile and execute the app. You will see the console application launch with a black box but because it doesn’t write anything to the console, nothing will be seen.
35. When the console app completes and the black box closes, check the changes you just made using the Term Store Management Tool:
    a) Open **Central Administration**:
       i) Start → All Programs → Microsoft SharePoint 2013 Products → SharePoint 2013 Central Administration
    b) Go to the **Term Store Management Tool** page:
       i) Application Management → Manage Service Applications → Managed Metadata Service
    c) In the list of service applications, select **Managed Metadata Service** to launch the **Term Store Management Tool**.
    d) Expand the **Example Taxonomies** group and find the terms you just modified:



In this exercise you worked with the SharePoint Managed client object model for taxonomies.