Hands-On Lab

Designing Data For Sharability

Lab version: 1.0.0

Last updated: 6/8/2015

Contents

[Overview 3](#_Toc338154805)

[Exercise 1: Adding Flip Tiles 4](#_Toc338154806)

[Task 1 – Adding and Replacing Assets 4](#_Toc338154807)

[Task 2 – Supporting Flip Tiles 5](#_Toc338154808)

[Task 3 – Testing Flip Tile Support 6](#_Toc338154809)

[Exercise 2: Adding Cyclic Tiles 12](#_Toc338154810)

[Task 1 – Supporting Cyclic Tiles 12](#_Toc338154811)

[Task 2 – Testing Cyclic Tile Support 14](#_Toc338154812)

[Exercise 3: Adding Iconic Tiles 19](#_Toc338154813)

[Task 1 – Changing the Manifest 19](#_Toc338154814)

[Task 2 – Supporting Icon Tiles 21](#_Toc338154815)

[Task 3 – Testing Iconic Tile Support 23](#_Toc338154816)

[Summary 27](#_Toc338154817)

Overview

* 1. With our basic chart complete, we will refine it visually and add valuable in-chart context to make our visualization as independent as possible, ready to be shared with the world.
  2. There are four types of live tiles in Windows Phone 8:
  3. **Standard tile**: The only type of tile available in Windows Phone 7.5, displaying an icon and title text. Use this type of tile for backwards compatibility. Unlike the other tiles, standard tiles do not support the wide tile size. Tiles created by pinning the app to the Start screen default to the standard tile if the developer does not provide another type of tile.
  4. **Cyclic tile**: A cyclic display of multiple images and text.
  5. **Flip tile**: An auto-flipping double-sided tile. In addition to the functionality of standard tiles, flip tiles support the wide tile format.
  6. **Iconic tile**: A static icon with a numeric badge indicating some application state. Wide iconic tiles can additionally display text content along a numeric badge.
  7. The developer can specify which tile type the app should create in different cases, as well as supply text andObjectives
  8. This lab provides instructions to achieve the following:
  + Supporting tiles from an app
  + Supporting the different types of live tiles (flip, cyclic, and iconic)

# Prerequisites

* 1. The following prerequisites are required to gain the most from this hands-on lab:
  + Microsoft Excel is required
  + Photoshop
    - Or Paint.NET (for PC) <http://www.dotpdn.com/downloads/pdn.html>
    - Or GIMP (for Mac or PC) <http://www.gimp.org/>

**Lab Structure**

This lab includes three exercises with the following tasks:

* + Finding and downloading a data set
  + Refining that data set
  + Creating a “rolling average” to smooth your data
  + Charting the data using Excel
  + Creating a chart snapshot and bringing it into an image manipulation program for visual refinement

# Estimated completion time

* 1. Completing this lab should take at least 60 minutes.

Exercise 1: Gather and prepare data

* 1. In this exercise we will modify the Contoso Cookbook app to support secondary tiles, starting with flip tiles. We will then test our changes and see how the new tiles enable direct navigation into the app, and display important information from the app on the Start screen.
  2. This exercise will concentrate on flip tiles; the subsequent exercises will demonstrate the two other tile types (cyclic and iconic).

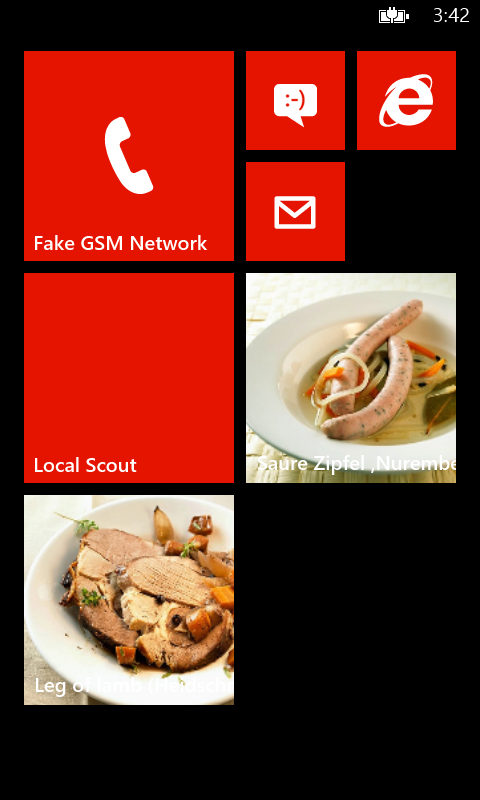
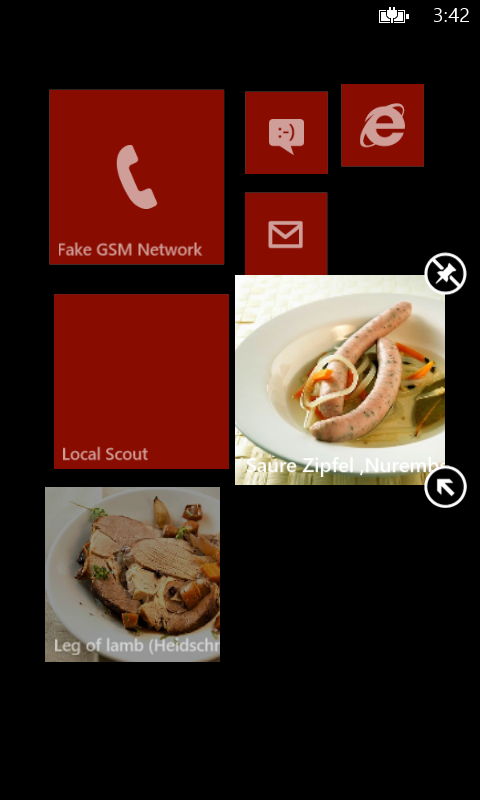
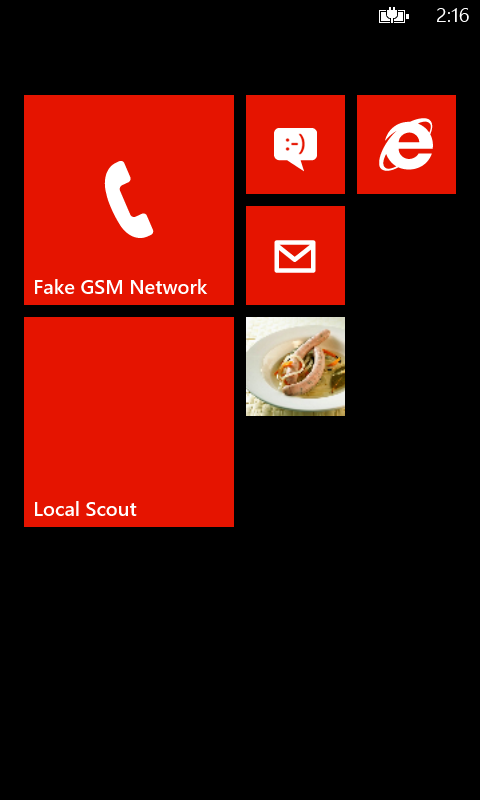
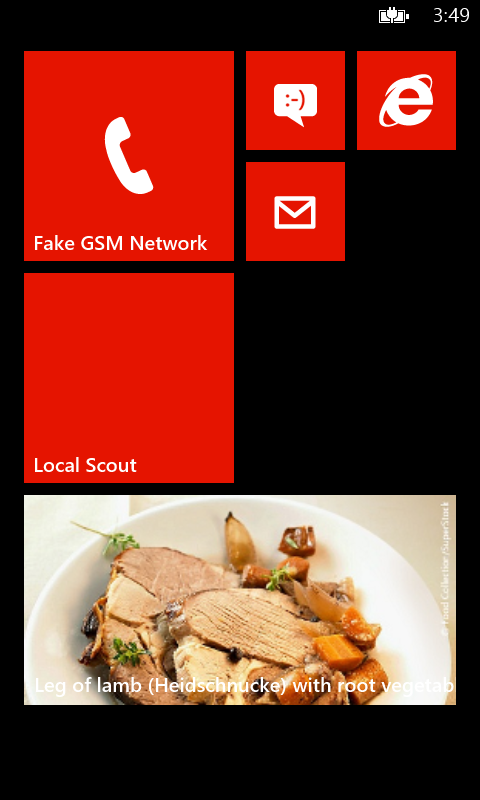
Task 1 – Adding and Replacing Assets

* 1. In this task we will add and replace the images needed to support tiles within our app.
  2. Open Visual Studio 2012.
  3. Navigate to the **EX1\Begin** folder.
  4. Open the **ContosoCookbook.sln** solution.
  5. Locate the **Assets** folder under the main project.
  6. Right-click the project in Visual Studio Solution Explorer and select Add > Existing item.
  7. Add the following image files from the **Assets** folder:
     1. MediumLogo.png
     2. SmallLogo.png
  8. Locate the **Assets/Icons** folder under the main project.
  9. Add the following image files from the **Assets/Icons** folder:
     1. like.png
     2. unlike.png
  10. These assets (images) are now loaded into our project, and will be used in the subsequent tasks.

Task 2 – Supporting Flip Tiles

* 1. To pin and unpin tiles, we have to replace and expand the functionality provided by the **Common.Features** class.
  2. Open the **Features.cs** file.
  3. Locate the **Tile** class.
  4. We will use a flip tile to display detailed recipe information on the back side of an automatically flipping tile. Locate and replace the contents of the **SetTile** method with the following code:
     1. C#
     2. FlipTileData tileData = new FlipTileData()
     3. {
     4. //Front square data
     5. Title = item.Title,
     6. BackgroundImage = new Uri(item.GetImageUri(), UriKind.Relative),
     7. SmallBackgroundImage = new Uri(item.GetImageUri(), UriKind.Relative),
     8. //Back square data
     9. BackTitle = item.Title,
     10. BackContent = MakeString(item.Ingredients),
     11. BackBackgroundImage = new Uri(item.Group.GetImageUri(), UriKind.Relative),
     12. //Wide tile data
     13. WideBackgroundImage = new Uri(item.GetImageUri(), UriKind.Relative),
     14. WideBackBackgroundImage = new Uri(item.Group.GetImageUri(),  
          UriKind.Relative),
     15. WideBackContent = item.Directions
     16. };
     17. ShellTile.Create(new Uri(navDataSource, UriKind.Relative), tileData, true);
     18. Previously, this method created a standard tile with the front showing the recipe image and the back showing a map of the country of origin. The new version creates a flip tile, which displays the ingredients when small- or medium-sized, and the recipe instructions when wide-sized.
     19. Note the **WideBackgroundImage**, **WideBackBackgroupImage**, and **WideBackContent** properties. By using these properties, we display more elaborate information on the tile front and back when it is wide.
  5. The preceding **SetTile** method uses the **MakeString** method to obtain the string representation of the recipe to be assigned to the **BackContent** property in wide tile mode. We now implement this method. Add the following method to the class:
     1. C#
     2. private static string MakeString(ObservableCollection<string> ingredients)
     3. {
     4. string res = "";
     5. foreach (var ingredient in ingredients)
     6. {
     7. res += ingredient + "\n";
     8. }
     9. return res;
     10. }
     11. Additionally, insert the following "using" directive at the top of the file:
     12. C#
     13. using System.Collections.ObjectModel;
     14. Note that the **SetTile** method is already called from the **btnPinToStart\_Click** event handler in **RecipeDetailPage.xaml.cs**, which requires no further changes. The result is that the flip tile will be used to display a secondary tile for each recipe that the user pins to the Start screen.

Task 3 – Testing Flip Tile Support

* 1. Our Contoso Cookbook app is now ready. We test the flip tile support by running the app:
  2. Compile, deploy, and run the application.
     1. 
     2. Figure 1
     3. Recipe Group Page
  3. Tap one of the recipe groups.
     1. 
     2. Figure 2
     3. German Recipe Group Page
  4. Scroll the pivot control to reveal the Recipes pivot item. If you see a blank screen with a "Buy this App" button, click the button to show the recipes.
     1. 
     2. Figure 3
     3. German Recipes
     4. Select a recipe.
     5. 
     6. Figure 4
     7. Saure Zipfel Recipe
  5. Tap the "…" button on the lower-right corner of the screen. The "pin" button appears, indicating that the recipe is not yet pinned to the Start screen.
  6. Tap the "pin" button and select "Yes" at the "Set as Lock Screen" prompt asking your approval for the app to become the lock screen background provider.
  7. Inspect the flip tile of your selected recipe toggling between the recipe's image on the front and the ingredients on the back.
  8. Tap the back button twice to return to the recipe group and pin another recipe in the same manner.
     1. Inspect both flip tiles on the Start screen.
     2. 
     3. Figure 5
     4. Flip Tiles on the Start Screen
     5. Tap and hold one of the tiles until the tile change icons appear.
     6. 
     7. Figure 6
     8. Editing a Flip Tile on the Start Screen
  9. Tap the "unpin" icon in the top-right corner of the tile. The tile should disappear.
  10. Tap and hold the other tile until the tile change icons appear.
  11. Tap the "Change size" icon in the bottom right corner of the tile repeatedly to toggle between the different tile sizes (small, medium, wide).
      1. 
      2. Figure 7
      3. Small Flip Tile on the Start Screen
      4. 
      5. Figure 8
      6. Wide Flip Tile on the Start Screen

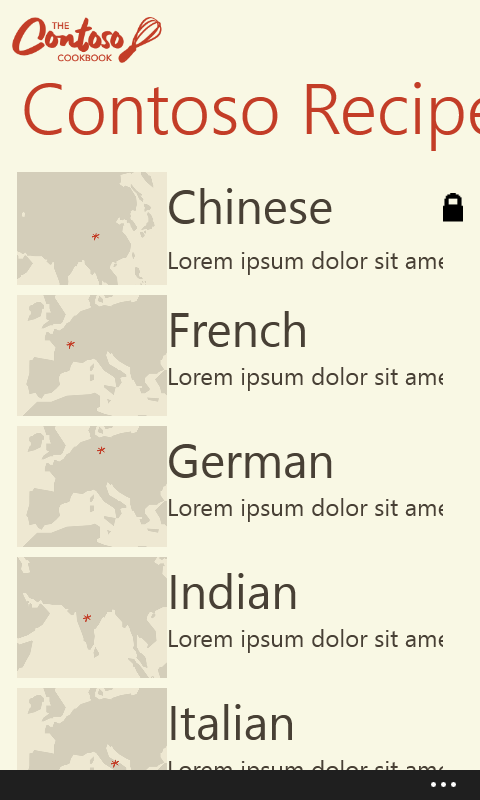
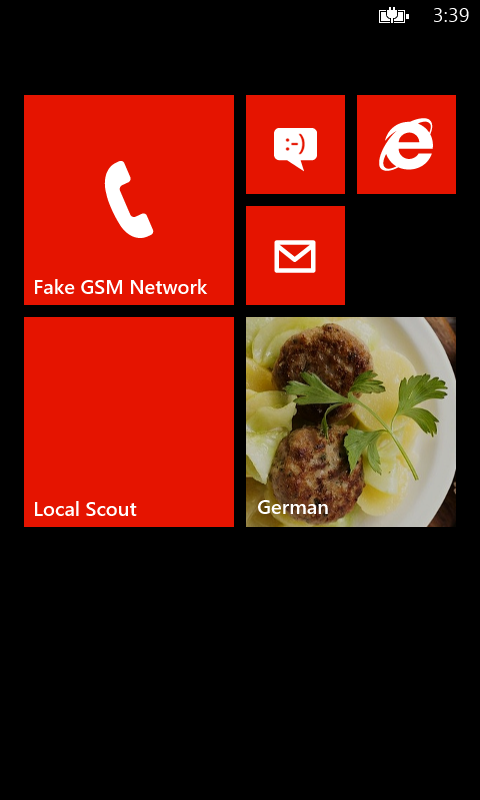
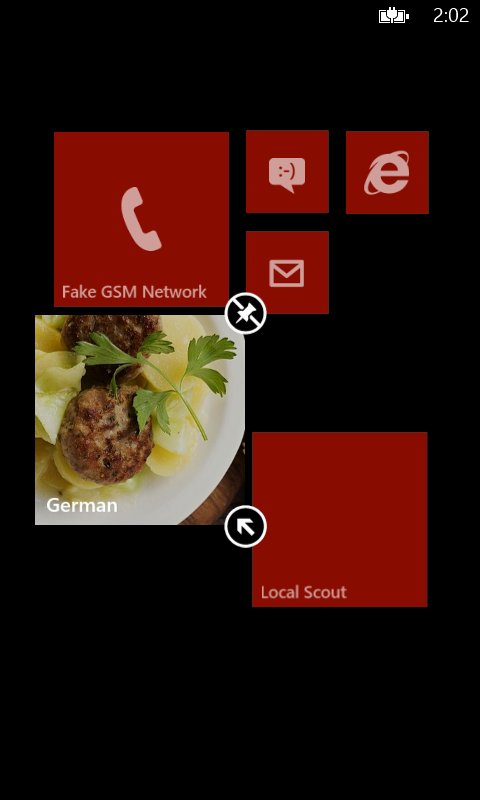
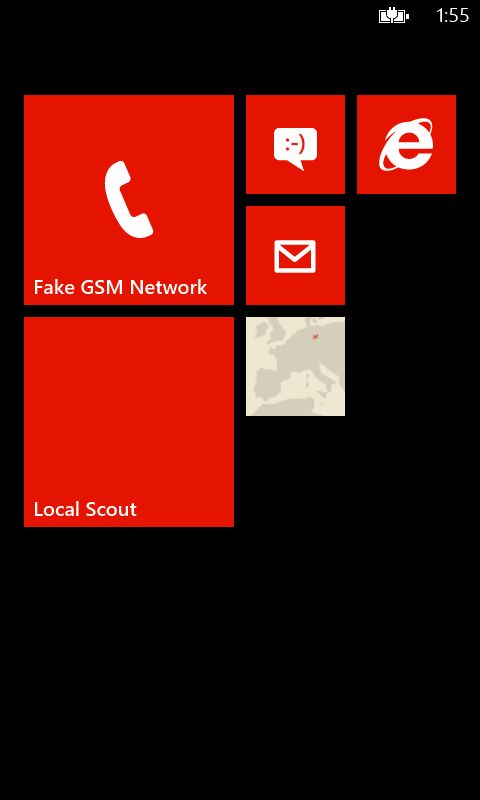
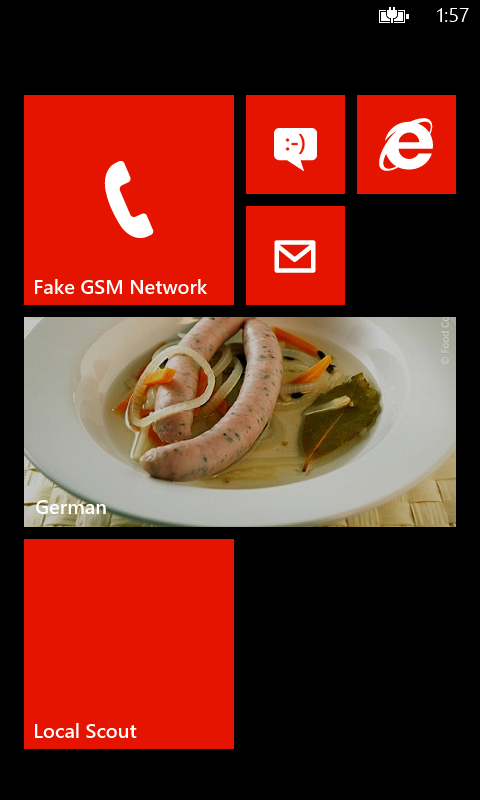
Exercise 2: Smooth and Chart Data

* 1. In this exercise we will add cyclic tile support to the Contoso Cookbook app.

Task 1 – Supporting Cyclic Tiles

* 1. A cyclic tile displays a repeating series of images. We will use a cyclic tile to display recipe images in a recipe group pinned by the user.
  2. As earlier, to support cyclic tiles, we will replace some of the common functionality provided by the **Common.Features** class.
  3. Open the **Features.cs** file.
  4. Locate the **Tile** class.
  5. Add the following new method:
     1. C#
     2. public static void SetGroupTile(RecipeDataGroup group, string navDataSource)
     3. {
     4. List<Uri> list = new List<Uri>();
     5. foreach (var recipe in group.Items)
     6. list.Add(new Uri(recipe.ImagePath.LocalPath, UriKind.Relative));
     7. CycleTileData tileData = new CycleTileData()
     8. {
     9. Title = group.Title,
     10. SmallBackgroundImage = new Uri(group.GetImageUri(),  
          UriKind.RelativeOrAbsolute),
     11. CycleImages = list
     12. };
     13. ShellTile.Create(new Uri(navDataSource, UriKind.Relative), tileData, true);
     14. }
     15. The preceding method creates a cyclic tile that flips through all the recipes in a recipe group. Configuring a cyclic tile involves setting the **Title**, **SmallBackgroundImage**, and **CycleImages** properties.
     16. Next, we add support for the user to pin recipe groups to the Start screen.
  6. Open the **GroupDetailPage.xaml** file.
  7. Locate and replace the **phone:PhoneApplicationPage.ApplicationBar** element with the following:
     1. XAML
     2. <phone:PhoneApplicationPage.ApplicationBar>
     3. <shell:ApplicationBar IsVisible="True" IsMenuEnabled="False"   
         Mode="Minimized" Opacity="1.0">
     4. <shell:ApplicationBarIconButton x:Name="btnPinToStart"   
         IconUri="/Assets/Icons/pin.png"  
         Click="btnPinToStart\_Click" Text="Pin To Start"/>
     5. </shell:ApplicationBar>
     6. </phone:PhoneApplicationPage.ApplicationBar>
  8. Open the **GroupDetailPage.xaml.cs** file.
  9. Add the following "using" directive at the top of the file:
     1. C#
     2. using Microsoft.Phone.Shell;
  10. Add the following constants to the **GroupDetailPage** class:
      1. C#
      2. private const string RemoveFavUri = "/Assets/Icons/unpin.png";
      3. private const string FavUri = "/Assets/Icons/pin.png";
      4. These constants will be used for the pin and unpin buttons.
  11. Add the following property:
      1. C#
      2. public ApplicationBarIconButton pinBtn
      3. {
      4. get
      5. {
      6. var appBar = (ApplicationBar)ApplicationBar;
      7. var count = appBar.Buttons.Count;
      8. for (var i = 0; i < count; i++)
      9. {
      10. ApplicationBarIconButton btn = appBar.Buttons[i]   
           as ApplicationBarIconButton;
      11. if (btn.IconUri.OriginalString.Contains("pin"))
      12. return btn;
      13. }
      14. return null;
      15. }
      16. }
  12. Next, we add a method that that modifies the button icon and text according to whether the recipe group is pinned to the Start screen. Add the following method:
      1. C#
      2. void SetPinBar()
      3. {
      4. var uri = NavigationService.Source.ToString();
      5. if (Features.Tile.TileExists(uri))
      6. {
      7. pinBtn.IconUri = new Uri(RemoveFavUri, UriKind.Relative);
      8. pinBtn.Text = "Unpin";
      9. }
      10. else
      11. {
      12. pinBtn.IconUri = new Uri(FavUri, UriKind.Relative);
      13. pinBtn.Text = "Pin";
      14. }
      15. }
  13. Locate the **OnNavigatedTo** method and replace the last line (the **base.OnNavigatedTo** method call), with the following code:
      1. C#
      2. SetPinBar();
      3. base.OnNavigatedTo(e);
  14. Finally, we should add an event handler for the pin and unpin buttons. If the user pins the group, we should create a new secondary tile; if the user unpins it, we should remove it. Add the following event handler:
      1. C#
      2. private void btnPinToStart\_Click(object sender, EventArgs e)
      3. {
      4. var uri = NavigationService.Source.ToString();
      5. if (Features.Tile.TileExists(uri))
      6. Features.Tile.DeleteTile(uri);
      7. else
      8. Features.Tile.SetGroupTile(group, uri);
      9. SetPinBar();
      10. }

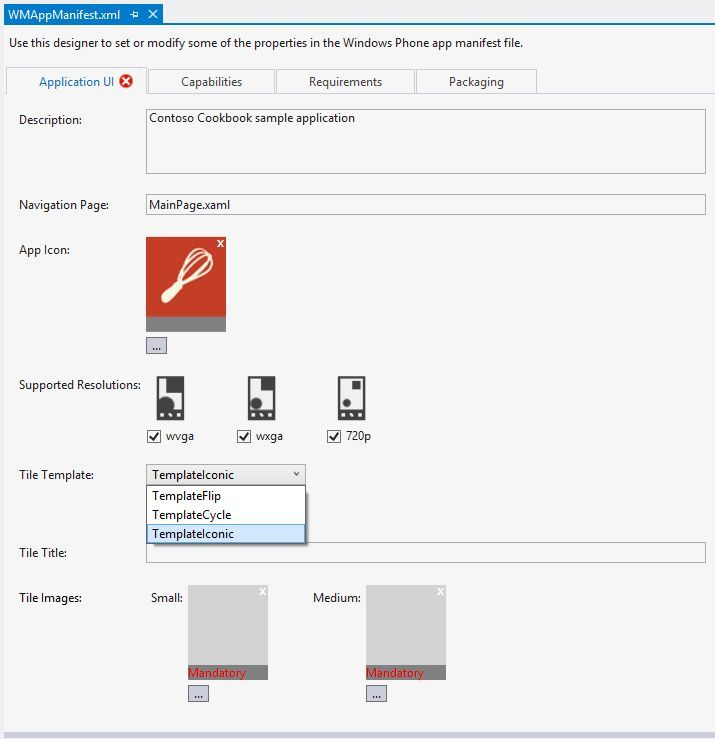
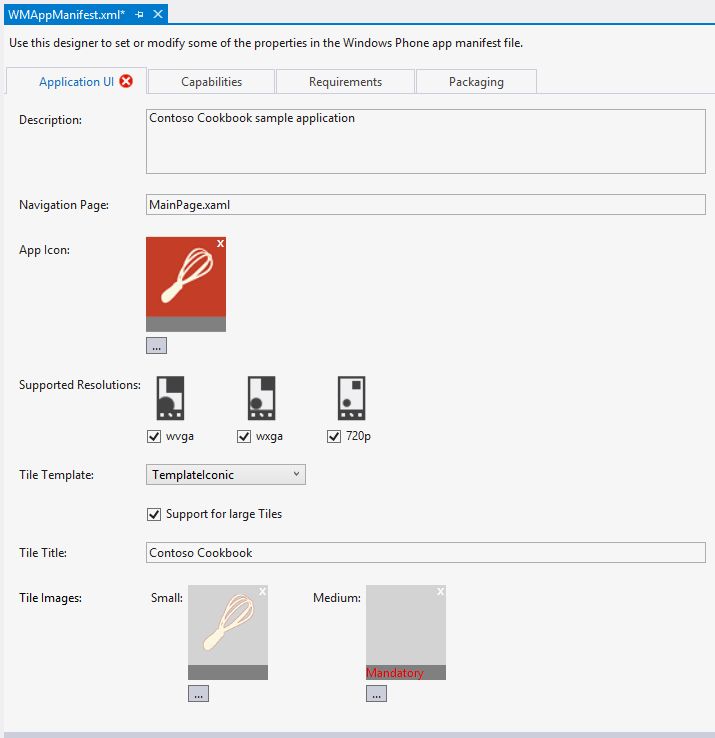
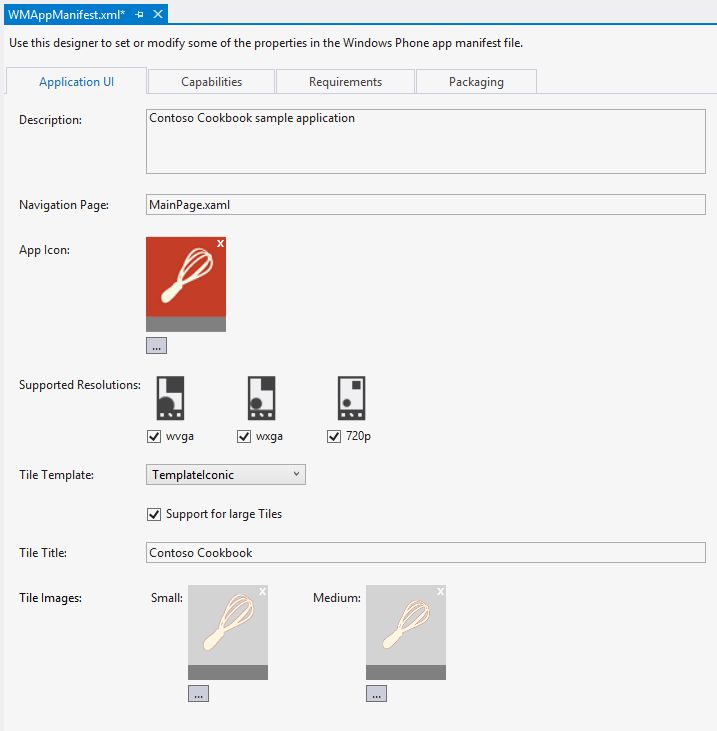
Task 2 – Testing Cyclic Tile Support

* 1. Our Contoso Cookbook app is now ready. We proceed to test the cyclic tile support by running it:
  2. Compile, deploy, and run the application.
     1. 
     2. Figure 9
     3. Recipe Group Page
  3. Tap one of the recipe groups.
     1. 
     2. Figure 10
     3. German Recipe Group Page
     4. Tap the "…" button on the lower-right corner of the screen. The “pin” button appears, because this recipe group was not pinned to the Start screen yet.
     5. 
     6. Figure 11
     7. German Recipe Group Page with Application Bar Visible
  4. Tap the "pin" button. The cyclic tile of the selected recipe group appears on the Start screen. Wait a few seconds to see the recipe images replace each other periodically.
     1. 
     2. Figure 12
     3. German Recipes in a Cyclic Tile Pinned to the Start Screen
  5. Tap and hold one of the tiles until the tile change icons appear.
     1. 
     2. Figure 13
     3. Editing a Cyclic Tile
  6. Tap the "Change size" icon in the bottom right corner of the tile repeatedly to toggle the various tile modes (small, medium, and wide).
     1. 
     2. Figure 14
     3. Small Cyclic Tile on the Start Screen
     4. 
     5. Figure 15
     6. Wide Cyclic Tile on the Start Screen

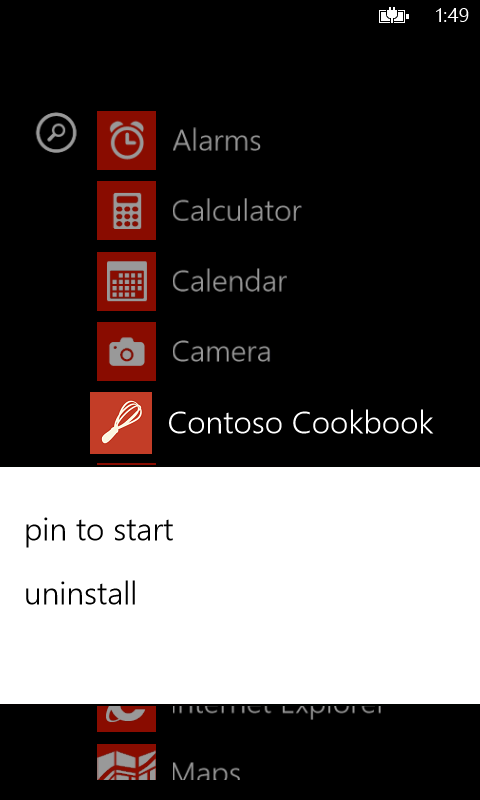
Exercise 3: Visually Refine Chart

* 1. In this exercise we will add iconic tiles to the Contoso Cookbook app.

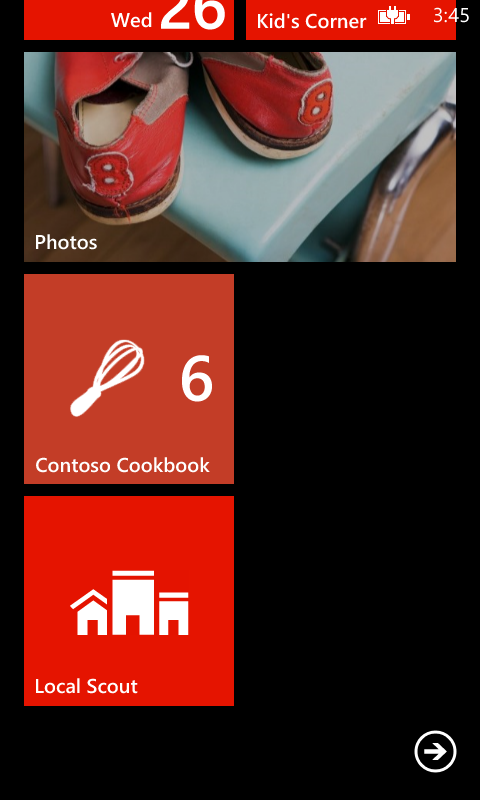
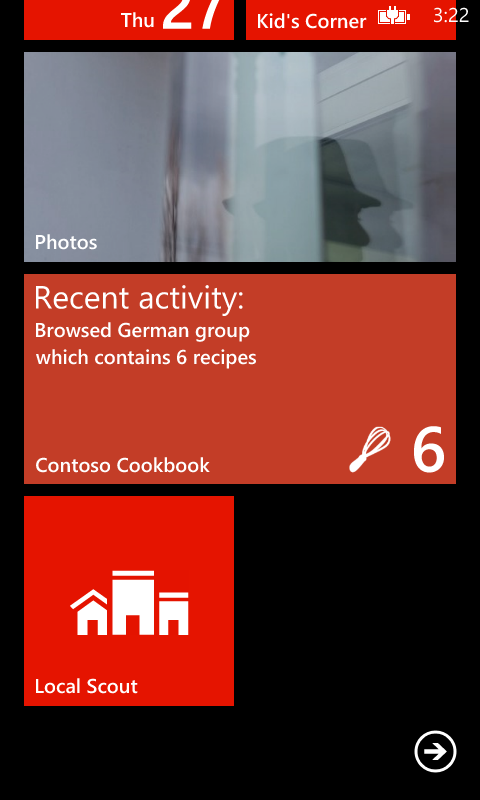
Task 1 – Changing the Manifest

* 1. First, we update our app’s tile template in the manifest to declare support for iconic tiles.
  2. Open the **WMAppManifest.xml** file.
  3. Go to the "Application UI" tab and scroll down to the end. In the "Tile Template" combo box, select "TemplateIconic":
     1. 
     2. Figure 16
     3. Selecting a Tile Template in the Application Manifest
  4. Type "Contoso Cookbook" into the "Tile Title" text field.
  5. Click the "…" button for the "Small" tile image and choose the **Assets\SmallLogo.png** file:
     1. 
     2. Figure 17
     3. Adding a Small Tile Image to the Application Manifest
  6. Click the "…" button for “Medium” tile image and choose the **Assets\MediumLogo.png** file:
     1. 
     2. Figure 18
     3. Adding a Medium Tile Image to the Application Manifest
  7. The application is now capable of displaying custom iconic tiles on the Start screen.

Task 2 – Supporting Icon Tiles

* 1. The Contoso Cookbook app will use iconic tiles to display brief information about the last visited recipe group on the app’s primary tile. Iconic tiles support a numeric badge, a title when medium- or wide-sized, and up to three lines of text in wide size.To pin our app’s primary tile to the start screen: scroll the Start screen left to reveal the list of installed apps, find the Contoso Cookbook app, tap and hold it, and select the "pin to start" option from the menu.
     1. 
     2. Figure 19
     3. Pinning the App to the Start Screen
     4. As before, to support iconic tiles, we replace some of the functionality provided by the **Common.Features** class.
  2. Open the **Features.cs** file.
  3. Locate the **Tile** class.
  4. Add the following “using” directive at the top of the file:
     1. C#
     2. using System.Windows.Media;
  5. Add the following new method:
     1. C#
     2. public static void UpdateMainTile(RecipeDataGroup group)
     3. {
     4. //Get application's main tile
     5. var mainTile = ShellTile.ActiveTiles.FirstOrDefault();
     6. if (null != mainTile)
     7. {
     8. IconicTileData tileData = new IconicTileData()
     9. {
     10. Count = group.RecipesCount,
     11. BackgroundColor = Color.FromArgb(255, 195, 61, 39),
     12. Title = "Contoso Cookbook",
     13. IconImage = new Uri("/Assets/MediumLogo.png",  
          UriKind.RelativeOrAbsolute),
     14. SmallIconImage = new Uri("/Assets/SmallLogo.png",
     15. UriKind.RelativeOrAbsolute),
     16. WideContent1 = "Recent activity:",
     17. WideContent2 = "Browsed " + group.Title + " group",
     18. WideContent3 = "which contains " + group.RecipesCount +   
          " recipes"
     19. };
     20. mainTile.Update(tileData);
     21. }
     22. }
     23. The preceding method updates the app’s primary tile, if it was not removed by the user, with an iconic tile. The tile displays a numeric badge indicating the number of recipes in the most recently visited recipe group. In wide tile mode, it also displays the name of the group and a more detailed description.
     24. Unlike the secondary tiles for recipe groups created in Exercise 2, the preceding code affects only the app’s primary tile, and should be updated whenever the user switches from one recipe group to another. We now add support for these updates.
  6. Open the **GroupDetailPage.xaml.cs** file.
  7. Locate the **OnNavigatedTo** method and replace the last line (the **base.OnNavigatedTo** method call), with the following code:
     1. C#
     2. //Update main tile with recently visited group
     3. Features.Tile.UpdateMainTile(group);
     4. base.OnNavigatedTo(e);
     5. The preceding code will update the app’s primary tile with the most recently visited group when the user navigates to it.

Task 3 – Testing Iconic Tile Support

* 1. Our Contoso Cookbook app is now ready. We proceed to test the iconic tile support by running it:
  2. Build, deploy, and run the application.
     1. 
     2. Figure 20
     3. Recipe Group Page
  3. Tap one of the recipe groups.
     1. 
     2. Figure 21
     3. German Recipe Group Page
  4. Tap the device’s home button. The app’s primary tile has changed to display information about the most recently visited group.
     1. 
     2. Figure 22
     3. Medium Primary Iconic Tile on the Start Screen
  5. Tap and hold the app’s primary tile until the tile change icons appear.
  6. Tap the "Change size" icon in the bottom right corner of the tile repeatedly to toggle between the various tile modes (small, medium, and wide). Note the detailed information that appears in wide mode.
     1. 
     2. Figure 23
     3. Small Iconic Tile on the Start Screen
     4. 
     5. Figure 24
     6. Wide Iconic Tile on the Start Screen

Summary

* 1. In this lab, you learned how to use tiles to supply information without forcing users to open your app. All tile types support three sizes (small, medium, and wide), and the user can toggle between the sizes to reconfigure the Start screen. By supporting various types of tiles – flip tiles, cyclic tiles, and iconic tiles – your app can offer a variety of pinned content on the phone’s Start screen.