**HOW DO I …**

Use push notifications with secondary tiles and deep toasts?

Greetings. In this video we will see how to incorporate some of Windows Phone Mango’s new push notification capabilities into an existing application. With Windows Phone Mango, applications may present several tiles on the phone’s start area and toast messages are capable of linking to specific application pages. Push notifications are now able to take advantage of these new features by updating individual application tiles and initiating deep toast messages. In this video, we will see how to alter a phone application and its corresponding server to make use of new push capabilities.

# Overview

Before we begin, let’s have a look at the application that we will be using throughout this video. We will use a simple client application that displays weather information for a predefined list of locations. The information is received using push notifications sent from an associated web service, and the service’s console can be used to send raw, tile or toast notifications.

It is important to note that all aspects of push notification handling are done using the push notification recipe, which you can find by following the link supplied on this slide.

# Push messages on Mango

With the release of Windows Phone Mango, push notifications are enhanced to allow leveraging some of Mango’s new features, such as multiple tiles for a single application, each featuring both a front and back side, and deep toast messages that link directly to a specific page in the application. To this end, the XML structure for tile and toast notifications has changed. Let’s have a look at a tile message.

# Tile message

This slide displays the new structure of a tile message. The attribute highlighted in blue is new and is used to indicate the tile which the message is directed at, using the tile’s navigation URI as an identifier. The elements highlighted in red are new as well and are used to supply information for the tile’s back side.

Next, let us have a look at the new structure for toast messages.

# Toast message

This slide displays the new structure of a toast message. The element highlighted in blue is new and is used to indicate the page to which the toast message should lead when clicked by the user.

Now that we are familiar with the new push capabilities, let us use them to enhance our application.

# Demo

Our task will be twofold, we will update the application to allow pinning and unpinning secondary tiles from the phone’s start area and will then enhance the application’s associated web service to send toast and tile notifications in the new form we have just presented.

1. Start by modifying the web-service and have it send deep toasts and messages directed at secondary tiles. Remember that the service must be run as administrator to work properly.
2. In the **WPPushNotification.ServerSideWeatherSimulator** project, open the **MainWindow.xaml.cs** file and navigate to the **sendToast** method. Change the “TODO” comment and both lines above it to the following:

toastPushNotificationMessage.Title = String.Format("WEATHER ALERT ({0})", cmbLocation.SelectedValue);

toastPushNotificationMessage.SubTitle = msg;

toastPushNotificationMessage.TargetPage = MakeTileUri(cmbLocation.SelectedValue.ToString()).ToString();

1. Move to the **sendTile** method and change the “TODO” comment there to the following line of code. It is important to remember that the change is so simple since we are using a version of the push notification recipe that is aimed at Windows Phone Mango:

tilePushNotificationMessage.SecondaryTile = MakeTileUri(location).ToString();

1. Go to the **RegistrationService\_DataRequested** method and change the “TODO” comment to the following line of code:

tilePushNotificationMessage.SecondaryTile = MakeTileUri(e.LocationName).ToString();

1. Add the following method to support the creation of navigation URIs from location names:

private static Uri MakeTileUri(string locationName)

{

    return new Uri(Uri.EscapeUriString(String.Format("/CityPage.xaml?location={0}",

        locationName)), UriKind.Relative);

}

1. Move to the **MainPage.xaml.cs** file under the **WPPushNotification.TestClient** project and fill the **UnpinItem\_Click** method with the following code:

LocationInformation locationInformation = (sender as MenuItem).DataContext as LocationInformation;

ShellTile tile = ShellTile.ActiveTiles.FirstOrDefault(

    t => t.NavigationUri.ToString().EndsWith(Uri.EscapeUriString(locationInformation.Name)));

if (tile == null)

{

    MessageBox.Show("Tile inconsistency detected. It is suggested that you restart the application.");

    return;

}

try

{

    tile.Delete();

    locationInformation.TilePinned = false;

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message, "Error deleting tile", MessageBoxButton.OK);

    return;

}

1. Add the following code to the **PinItem\_Click** method:

LocationInformation locationInformation = (sender as MenuItem).DataContext as LocationInformation;

Uri tileUri = MakeTileUri(locationInformation);

StandardTileData initialData = new StandardTileData()

{

    BackgroundImage = new Uri("Images/Clear.png", UriKind.Relative),

    Title = locationInformation.Name

};

((sender as MenuItem).Parent as ContextMenu).IsOpen = false;

try

{

    ShellTile.Create(tileUri, initialData);

}

catch (Exception ex)

{

    MessageBox.Show(ex.Message, "Error creating tile", MessageBoxButton.OK);

    return;

}

1. Move to the **ChangeMainTile\_Click** method and add some additional code to allow manually updating the application’s main tile:

// Get the main tile (it will always be available, even if not pinned)

ShellTile mainTile = ShellTile.ActiveTiles.FirstOrDefault(t => t.NavigationUri.ToString() == "/");

StandardTileData newData = new StandardTileData()

{

    BackgroundImage = new Uri(String.Format("Images/MainTile/{0}.png", (listMainTileImage.SelectedItem as ListPickerItem).Content), UriKind.Relative),

    Title = txtMainTileTitle.Text

};

mainTile.Update(newData);

1. Finally, go to **App.xaml.cs** and fill out the **RefreshTilesPinState** method, which updates the pin state of each location according to whether or not its tile is available, with the following code:

Dictionary<string, LocationInformation> updateDictionary = Locations.Values.ToDictionary(li => li.Name);

foreach (ShellTile tile in ShellTile.ActiveTiles)

{

    string[] querySplit = tile.NavigationUri.ToString().Split('=');

    if (querySplit.Count() != 2)

    {

        continue;

    }

    string locationName = Uri.UnescapeDataString(querySplit[1]);

    updateDictionary[locationName].TilePinned = true;

    updateDictionary.Remove(locationName);

}

foreach (LocationInformation locationInformation in updateDictionary.Values)

{

    locationInformation.TilePinned = false;

}

# Summary

In this video, we presented the push notification enhancements that are part of Windows Phone Mango and saw how to use them to enhance an existing application’s functionality.