**HOW DO I …**

Add a background agent to my application?

This is the second of two videos. In the first video, we saw how to take an application and enhance it by having the application support secondary tiles. In this video, we will see how to further enhance the experience by creating a background agent to update the tiles even while the application is not active.

# Overview

In case it has been a long time since you saw the first video, or if you skipped it entirely, let us do a quick review of the application we will be working with. Wazup is a Silverlight application which allows the user to keep track of several online services: The Windows Phone 7 blog, Digg and Twitter. In the previous video we added the ability to pin specific tweet trends as secondary tiles, so that the user may examine them more conveniently. Let’s have a look at the actual code.

We can enhance our application by updating our secondary tiles with trend-related information and to do this we will use one of Mango’s new multi-tasking features.

# Mango Multitasking Support

New to Windows Phone Mango is the ability for applications to perform background operations, effectively enabling multi-tasking. Mango’s multi-tasking model aims to be healthy for the device, while still presenting the user with an integrated and delightful experience. To this end, background operations may only be performed in one of several defined manners.

We will be focusing on periodic background agents, as we wish to add one to our application.

# Periodic background agents

Periodic background agents are meant to perform small tasks that take no longer than 15 seconds, which is the amount of time they are allowed to run, and are invoked every half an hour. While working, the agents need to take care not to strain the system too much, and constraints are in place to prevent such excessive strain.

An application registers its agent while it is active, and it proceeds to run while the application itself is inactive. Each application is limited to a single periodic agent and it needs to renew its registration with the device, lest it become inactive after 14 days.

In our application, we can add a periodic background agent to update the amount of new tweets available in each of our pinned tiles, as well as notify us of extraordinarily interesting tweets. Let’s see how this is done:

1. Start by adding a new project to the solution. Add a “Windows phone scheduled task agent” project and name it **TrendSamplingAgent**. Add a reference to **WazzupCommon** in this new project.
2. Add the following using directives in **ScheduledAgent.cs**:

using Wazup.Helpers;

using System;

using Microsoft.Phone.Shell;

using Wazup.Services;

using System.IO.IsolatedStorage;

using System.Threading;

using System.Linq;

1. Add the following constant to the agent class:

private const string specialToken = "cake";

1. Add the following code to the top of the agent’s **OnInvoke** method. This code implements the agent’s operation. (Since it is quite a bit of code it is important to review it with the user, emphasizing the need for synchronization code due to the independent requests performed for each of the secondary tiles present):

string lastTokenTrendName = null;

int finishedRequestCount = 0;

int secondaryTileCount = ShellTile.ActiveTiles.Count() - 1;

ManualResetEvent requestsFinishedEvent = new ManualResetEvent(false);

// Go over each of the pinned tiles and update them

foreach (ShellTile activeTile in ShellTile.ActiveTiles)

{

    string tileTrendName = UriHelper.GetTrendNameFromUri(activeTile.NavigationUri);

    if (!IsolatedStorageSettings.ApplicationSettings.Contains(tileTrendName))

    {

        // This will happen for the main tile

        continue;

    }

    ShellTile tileToUpdate = activeTile;

    Trend trend = (Trend)IsolatedStorageSettings.ApplicationSettings[tileTrendName];

    // Get the tweets for the trend associated with the current tile

    TwitterService.Search(trend.name, tweets =>

        {

            int newTweetCount = 0;

            // See which tweets are new and/or contain a specific token and update the tile accordingly

            foreach (Tweet tweet in tweets)

            {

                if (tweet.created\_at > trend.last\_tweet\_fetch)

                {

                    newTweetCount++;

                }

                if (tweet.text.Contains(specialToken))

                {

                    lastTokenTrendName = trend.name;

                }

            }

            trend.last\_tweet\_fetch = DateTime.Now;

            StandardTileData newTileData = new StandardTileData

            {

                BackContent = "Updated - " + trend.last\_tweet\_fetch.ToShortTimeString(),

                Count = newTweetCount,

                BackTitle = trend.name,

                Title = trend.name

            };

            tileToUpdate.Update(newTileData);

            if (Interlocked.Increment(ref finishedRequestCount) == secondaryTileCount)

            {

                // Signal that all tweet updates are complete

                requestsFinishedEvent.Set();

            }

        });

}

requestsFinishedEvent.WaitOne();

// If we found a special token in one of the tweets, pop a toast message

if (lastTokenTrendName != null)

{

    ShellToast toastMessage = new ShellToast

    {

        Title = String.Format("There is {0} in one of your tweets!", specialToken),

        Content = String.Empty,

        NavigationUri = UriHelper.MakeTrendUri(lastTokenTrendName)

    };

    toastMessage.Show();

}

1. Add a reference to our new background agent project in the main **Wazup** project and see how the application’s manifest is automatically changed.
2. Open **App.xaml.cs** and add the following using directive:

using Microsoft.Phone.Scheduler;

1. Add the following constant to the **App** class:

public const string TrendTaskName = "TrendTask";

1. Create a new helper method named **CleanIsolatedStorage** andplace the current code from the **Application\_Launching** handler inside it.
2. Add an additional helper method for registering the application’s background agent:

private void SetUpBackgroundTask()

{

    ScheduledAction trendUpdateTask = ScheduledActionService.Find(TrendTaskName);

    if (ScheduledActionService.Find(TrendTaskName) == null)

    {

        trendUpdateTask = new PeriodicTask(TrendTaskName)

        {

            Description = "Looks for new tweets in pinned trend tiles.",

        };

        ScheduledActionService.Add(trendUpdateTask);

    }

    else if (trendUpdateTask.IsEnabled)

    {

        // Refresh the background task's scheduling

        ScheduledActionService.Remove(TrendTaskName);

        ScheduledActionService.Add(trendUpdateTask);

    }

}

1. Now simply call the methods added in the two previous steps from the **Application\_Launching** handler :

private void Application\_Launching(object sender, LaunchingEventArgs e)

{

    CleanIsolatedStorage();

    SetUpBackgroundTask();

}

1. Run the application and see how the background agent now appears in the agent management screen.
2. Add some code to allow for convenient debugging of the agent. Open **App.xaml.cs** and add the following code to the very end of the **SetUpBackgroundTask** method:

#if DEBUG

         ScheduledActionService.LaunchForTest(App.TrendTaskName, TimeSpan.FromMinute(1));

#endif

1. See how the debugger now attaches to the agent’s code after waiting for a minute.

# Summary

In this last of two videos we learned about part of Windows Phone Mango’s new multi-tasking capabilities by adding a periodic background agent to enhance our application. Updating secondary tiles is just one of many ways that multi-tasking can be leveraged while creating applications and you are encouraged to explore other possibilities.