Session 11 – Background tasks demos

# Demo 1 – Basics of Registering a Background Task

This is a simple solution. 2 projects. One is a background task (IBackgroundTask) as a WinRT component written in .NET and the other is just a simple blank project (single page XAML app).

1. Start by showing the code in the background task. All it does is it async delays around a 20 second period publishing progress to any interested listeners (i.e. the foreground app if it’s there) and then it writes the last time it ran to a file.
2. In the foreground app project, open the manifest and show that it has a “System event” background task specified on the Declarations section.
3. Show the code in MainPage.xaml.cs. There’s code in there for;

* Registering the background task using a SystemTrigger with the time zone change event.
* Unregistering it.
* Getting hold of it if it’s already registered.
* Sync’ing up event handlers to the background task’s progress/completed events.
* Updating the few bits of UI accordingly.

1. Run the app. On first run, the app will display UI to say that the task isn’t registered and a button to register it.
   1. Click the Register button. Once registered, the app will display UI to show the current status (running/not), the current progress (1->10), the last time the background task ran and a button to unregister it.
2. Note that the app registers a SystemTrigger using the old faithful time zone change event.
   1. So, to fire the trigger you must **first use the debug location toolbar to Suspend the app**, (if you don’t do this, the app is still running under the debugger, so when you change the time zone, the Progress event handler in the foreground app executes and you get an exception when it tries to update the UI)
   2. then go into the settings UI and change the time zone to something an hour or two away from the current zone.
   3. The background task will fire and if you then use the Debug location toolbar to Resume the foreground app you will see the Progress count will update in the foreground app.
3. Optional: Could update the background task to show a Toast when it runs. Then close the foreground app and change the TimeZone again = toast will show.
4. Run the app and then use the debugger to invoke the background task so we avoid any delays waiting for see something happen.

# Demo 2 – Location Trigger

There’s a very simple UI here. There’s a button which will set up a 500m geofence around a particular lat/lon with a particular ID.

The UI is hard-coded to make it easy to set up a fence called “microsoft” around Building 10 in Redmond.

We can run the app, add the fence and then register the background task. Once the task is registered it should trigger as the device enters/exits the geofence.

1. Run the app.
2. Click the Add 0.5km Geofence button.
3. Before registering the background task, open the location tab in the emulator and;
   1. Change the accuracy profile to (Urban) – this might be a red herring.
   2. Enter “microsoft, redmond” in the search box to find MS campus.
   3. Select Route in the dropdown
   4. Click the Load recorded location data from a file button (to left of Route dropdown) – open Locations.xml in the root folder of this demo app
4. Back in the app, click the button to Register the background task.
5. Back in the emulator location simulator, select Biking in the right-hand dropdown box, then click ‘Play’ button – the route plays as if you are riding a bike.
6. Wait for a geofence trigger to fire – it can take a little while. Battery optimizations can mean that events aren’t delivered more frequently than every 2 minutes so you might have to wait a while.

When a geofence does fire, there should be a toast notification and then an entry should be added to the list on the screen.

Notes on the project: there’s a class library which is shared by the foreground app and the background task and that library essentially stores/retrieves the fence notifications into a little file – this is how foreground/background communicate with the foreground app also handling the Completed event on the background task which is how it knows when to re-read the file and get the latest notifications.