Agent Team Lite Window Phone 7 Apps

By Boua Lor

Professor Advisor: Munehiro Fukuda

Prerequisites

Here are the programs needed to develop in window phone 7.

1. Visual Studio 2010 (any version)
2. Windows Phone Developer Tools RTW
3. Windows Identity Foundation Runtime
4. Silverlight 4 Tools for Visual Studio
5. Internet Information Services (IIS) with ASP.NET and IIS 6 compatibility enabled.
6. The Agent Team Lite Solution—Given by Fukuda

Agent Team Lite Solution File

Agent Team Lite Items

There are a few different layouts for window phone 7: the portrait, landscape, user control, panorama, and pivot page. The portrait page was mainly used in the application along with one panorama page. A panorama page allows multiple contents to be in one application. In addition, a portrait page when the phone is vertical and upright with one page of content.

The reference uses functionality of the services provided by the platform or other libraries. In order to use the methods, the application needs to reference it in the xaml file they will be using. In the solution, one addition reference was downloaded and used. You must **ADD reference:** Microsoft.Phone.Controls to allow the user to move right and left to navigate to another page.

The solution files contain many items and classes that are associated with each other. For example, for every xaml file, there is a cs too. The xaml file is the user interface and the cs file defines the action or event associated with the xaml file. Here is a list of all the xaml and cs file in the application and their functionality.

App–is the entry point of the application. It initializes resources and displays the application to the user.

MainPage—defines the page with the user interface of the application for the launch menu. It is the main page of the Agent Team Lite, it contains the main menu screen to create and view a job.

Connector—defines the page with the user interface of the application for the connector. It allows the user to choose the connector with the sensor map, sensor id, etc.

JobExe – defines the page with the user interface of the application for the job execution. It allows the user to create a job by selecting program name, arguments, filemap, etc.

JobHistory— defines the page with the user interface of the application for the job history. It allows the users to see their job history they have created.

JobStatus— defines the page with the user interface of the application for the job status. It display to the users if a job has been completed and how many jobs are running.

MachineSetup— defines the page with the user interface of the application for the machine setup. It displays the nodes that are currently running in the machine. Users may choose to stop or start the nodes.

Results— define the page with the user interface of the application for the job results. It displays the job results with a graphic, job status, etc.

SignIn— defines the page with the user interface of the application for the users to sign in and read the xml file of the site.

Agent Team Lite Current Implementation and Future Requirement

MainPage

Current:

- The users are able to view and click on "Create Job," "Job Status," "Job History" and "Sign In." These links shall take the users to the correct pages corresponding to the titles.

Future:

- A personalize background shall be added.
- Completed, unless other items need to be added to the launching page.

MachineSetup

Current:

- The checkbox represents the nodes that are available to the users to stop and start. There is a stop and start button for each checkboxes. The checkbox and buttons has no functionality at this point.
- The users may chose to go back to launch menu by swiping left and go forward by swiping right to the connector setup on the device.
- The save button moves the users to the connector page.

Future:

- For every node available to the users, there shall be a checkbox to represent it. When the checkbox is click on and the start button is pressed, the server shall recognize the call and start the node. When the checkbox is click on and the stop button is pressed, the server shall recognize the call and stop the node.
- The users shall save their decision by clicking on the save button or by swiping right to the next step.
- A personalize background shall be added.

Connector

Current:

- The users are able to input into the textboxes provided. Whatever the user input into the textboxes would show up at the bottom half of the page. This page instructs the developers how to extract information from the textboxes so they may be able to use the arguments inside the textboxes to make functions calls in the future. There is no connection to the webserver at this point.
- Aradio button is implemented. When the button is click, the textbox field will not take any texts from the users because the users are indicating that they want to use the defaulted setting.
- The users may chose to go back to launch menu by swiping left and go forward by swiping right to the connector setup on the device.

Future:

- Once the webserver has been connected to the application, the connector page shall allow the users to setup the sensor map, sensor id, and GUI. All the textbox argument shall be taken by the application and make the correct function call to the webserver.
- The radio button shall toggle between clicks. For example, if the radio button is clicked and the textboxes are disable another click would enable the textboxes (I have not successfully found the logic for this radio button toggling yet).
- If the Agent Team Lite team can provide some valid information of the sensor map, sensor id, and GUI the application can do some error checking.
- The users shall save their decision by clicking on the save button or by swiping right to the next step.
- A personalize background shall be added.

JobExe

Current:

- The users are able to input into the textboxes provided. There is no connection to the webserver at this point.
- The application currently does some error checking for the numbers field such as the CPU, memory, disk, and network.
- The run button triggers the logic to check the number field to ensure they are within range.
- The users may chose to go back to launch menu by swiping left and go forward by swiping right to the connector setup on the device.

Future:

- Once the webserver has been connected to the application, the job execution page shall execute a job based on the textbox fields. All the textbox argument shall be taken by the application and make the correct function call to the webserver.
- The users shall save their decision by clicking on the save button or by swiping right to the next step.
- If the Agent Team Lite team can provide some valid information to compare with the program name, filemap, and arguments, it can ensure a valid job will always be run.
- A personalize background shall be added.

JobHistory

Current:

- It is currently a blank page.

Future:

- Once the webserver has been connected to the application, the job history page shall pull the xml file from the website and display the job history to the users. Depending on the users, the application can display all the users' results or just his results.
- A personalize background shall be added.

JobStatus

Current:

- It is currently has a result link. This is for a job that can still be running.
- When the result link is click on, it displays the results page.

Future:

- Once the webserver has been connected to the application, the job status page shall pull the xml file from the website and display the all the jobs that are still running.
- When the users click on the one of the job results, the results page shall appear.
- A personalize background shall be added.

Results

Current:

- Titled as the Monitoring Panel, this page contains three pages: Job Status, Connector Console, and Graphic.
- Each page is connect to a browser from the internet. The Job Status is currently connected to www.google.com, the Connector Console is connected to www.connect.com, and graphics is connected to www.bing.com.

Future:

- Once the webserver has been established, the result page shall allow the users to interact with their jobs through the web browser. For example, through the job status, the users shall be able to see the node execution and through the graphics, the users can see the graphics that is posted online.
- A personalize background shall be added.

SignIn

Current:

- http://blogs.msdn.com/b/alikl/archive/2010/11/13/windows-phone-7-and-restful-services-delegated-access-using-azure-appfabric-access-control-service-acs-and-oauth.aspx this is a good website to connect to a RESTful service on window phone 7.
- This page is purely to test the website above about connecting to a restful service. There are many instructions so please read carefully. According to my research, this is the best way unless we want to create a webserver on Azule, which would make this much easier!
- You must create your own account on azule to be able to use this service.
- You must **ADD reference**: SL.Phone.Federation.Controls for the code to work.
- It will let you sign into google or Hotmail. However, it will return error after you login.

Future:

- The application shall allow the users to sign in with your UW netid.
- The application shall authenticate the users before launching the main menu.

Images and Icons:

Current/Future:

An icon and background has been created for this application. However, whenever the new background or icon is added into the code, the code breaks and it stops working. Therefore, the background and icon images shall remain in the "Images" folder until someone can successfully add them without breaking the code. The images names are bg.png and iconImage.png.

Conclusion

There are still many things that need to be implemented and connected for the window phone 7 Agent Team Lite's application once the server has been established. Although most of the UI work is completed, the application is still not done.