New Developer / Machine setup

Author: Jon Totton

Date: August 1, 2013

# Prerequisites

* Create a github account.
  + Clone the forerunner repository
* Ask to have a new ForerunnerSw / Office 365 account setup.
  + Install Lync and setup your desktop environment as per the Office365 instructions
* Ask to be added to the MSDN account.
* Ask to be added to Jira (I.e., atlassian.com)

# Machine Setup (applications / tools)

* Login and go to the MSDN subscriptions download page [here](https://msdn.microsoft.com/subscriptions/securedownloads/).
* Download and install "Visual Studio Ultimate 2010".
* Download and install "Visual Studio Ultimate 2012".
  + Note that you can open and install directly from the .iso file.
* Download and install "SQL Server 2012 Enterprise Edition with Service Pack 1 (x64)"
  + Setup choices:
    - I accepted the default product key. My understanding is that this key is just a 180 day trial version. So in the future, a new key will be needed (or a shared SQLServer could be used I suppose).
      * You will most likely get a warning about the Windows Firewall not having the appropriate ports open. You can find a page [here](http://go.microsoft.com/fwlink/?LinkId=94001) that describes what to do to open the ports you may need. You will not need to open ports unless you need to enable external devices to view the DB. Debugging via Visual Studio, locally, will be fine without opening any ports.
    - In the "Setup Role" screen I selected the "All Features With Defaults" option.
    - I then selected the default choices for all configuration settings.
* Download and install Git Extensions
  + <http://code.google.com/p/gitextensions/>
  + Setup choices:
    - I chose "puTTY" for the SSH Client choice

# Northwind DB and Test Report Configuration:

* Restore the Northwind database
  + Copy the file: C:\Users\Jon\Documents\GitHub\Forerunner\Test\Northwind.bak to the default SQL Server backup folder (E.g., C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup)
  + Use the SQL Server Management Studio to import the DB
* Assign the SSRS "Content Manager" role to your windows account. There are two ways that have worked to set this up. If you have trouble with the first way, try the second as follows:
  1. Assign SSRS roles to your windows account running with administrator privileges
     + Run IE "as administrator"
     + Launch the Report Manager: <http://localhost/Reports>
     + Select the "Folder Settings" -> "New Role Assignment"
     + Assign the SSRS "content Manager" role to your windows account
  2. Assign SSRS roles to your windows account running with administrator privileges using the Administrator user
     + Run CMD as administrator (right click run as administrator)
     + Type “net user administrator /active:yes” in the cmd window
     + Set the password for the administrator account, it will now show up
     + You can now login as the administrator
     + Run IE as administrator while logged in as the administrator
     + Go to Report Manager, folder settings and add your normal account as a content manager.
* Deploy the test reports
  + Make sure you have SQL Server Data Tools (SSDT) for 2010 installed
  + Open the VS 2010 SSDT project: Forerunner\Test\Test Reports 2012\Report Project1.sln
  + Right click the "Northwind Test Reports" project and make sure the "TargetServerURL" properly matches your machine setup (I.e., the port number may have to change) in the property pages dialog
  + Right click the "Northwind Test Reports" again and deploy the reports
  + Create the stored procedures needed by the reports:
    - Open SQL Server Management Studio
      * Select the Northwind DB
      * Open a new Query window
      * Copy the content from the file: …\GitHub\Forerunner\Test\Test Reports 2012\SimpleStoreProcedure.txt into the query window and execute the query.
  + At this point you should be able to use the SSRS Report Manager and Report viewer
    - Verify that you can see the Northwind DB and the deployed reports

# Forerunner Report Manager Configuration:

* Assign permissions for external users to have access to the IP / Port you will be using via the following command:

netsh http add urlacl url=http://192.168.1.20:9000/ user=everyone

* Edit the file under the IIS Express folder and add the IP address as a binding

File: …\Documents\IISExpress\config\applicationhost.config

Example:

<site name="ReportManager" id="2">

<application path="/" applicationPool="Clr4IntegratedAppPool">

<virtualDirectory path="/" physicalPath="C:\Users\Jon\Documents\GitHub\Forerunner\RS\Reporting\ReportManager\ReportManagerMVC\ReportManager" />

</application>

<bindings>

<binding protocol="http" bindingInformation="\*:9000:localhost" />

<binding protocol="http" bindingInformation="\*:9000:192.168.1.126" />

</bindings>

</site>

To test remotely from a device you may need to open your firewall.

In the same file you need to change the following:

<security>

<authentication>

<anonymousAuthentication enabled="false" userName="" />

<windowsAuthentication enabled="true">

<providers>

<add value="Negotiate" />

<add value="NTLM" />

</providers>

</windowsAuthentication>

* Open the solution:

Forerunner\RS\Reporting\ReportManager\ReportManagerMVC\ReportManagerMVC.sln in Visual Studio 2012

Set the configuration parameters; open the web.config file and change all references to server names to match your machine configuration(s), examples:

<add key="Forerunner.ReportServerWSUrl" value="http://localhost/ReportServer" />

<add key="Forerunner.ReportServerDBDomain" value="jonto-i7" />

For NTLM authentication; make sure the following lines are set properly:

<add key="Forerunner.UseIntegratedSecurityForSQL" value="false" />

* Create and provision a local windows test account
  + Create a windows user in the local domain with the values of the keys: ForeRunner.TestAccount and ForeRunner.TestAccountPWD.
  + Assign the "Browser and content manager" roles to the TestAccount user

* At this point you should be able to run the report Manager in VS 2012 and load the Report Manager into the browser with the URL: <http://localhost:9000>

# Install the emulators

* Install the Windows Phone SDK 8.0 from here:

<http://www.microsoft.com/en-us/download/details.aspx?id=35471>

* Install iPhone emulators running on windows

Note that the recommended way to debug on iOS devices is by using safari on a MAC computer using the “Developer” capabilities. You can use either the iOS simulator or a physical device directly connected via USB with Safari. You may get some benefit from the windows emulators but I found them very limited. You can find a couple I found here:

<http://visualstudiogallery.msdn.microsoft.com/6bed5adb-1d6a-483d-9e22-ae0f88ec4477>

Or

<http://www.browserstack.com/>

* Install Android emulators here:

<http://developer.android.com/sdk/index.html>

* Install Blackberry emulators here:

<https://developer.blackberry.com/devzone/develop/simulator/sim_index.html>