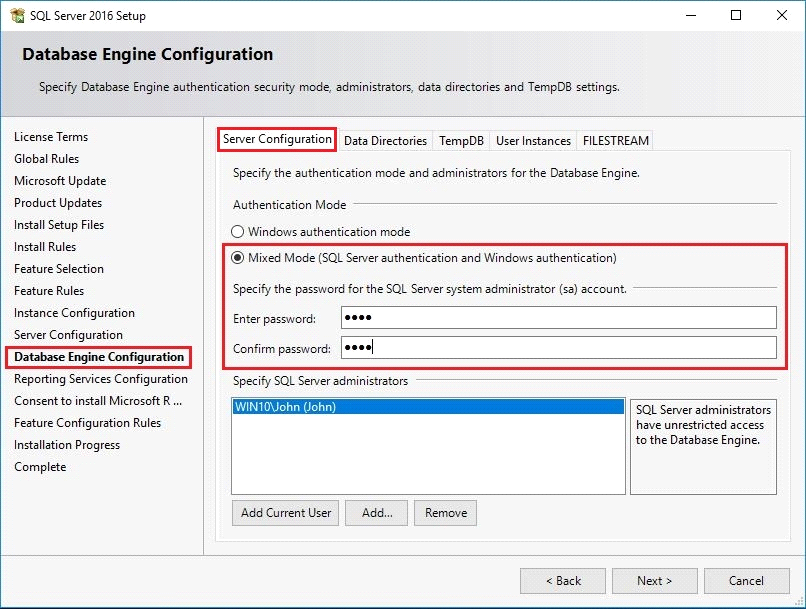
***R2G2 Post Windows Installation Setup (Development)***

Windows Programs and Features

* Ensure the following **Windows Features** are installed:
* Internet Information Services
* ASP.NET
* .Net Extensibility
* .Net 4.6.1 Extensibility
* Windows Message Queuing

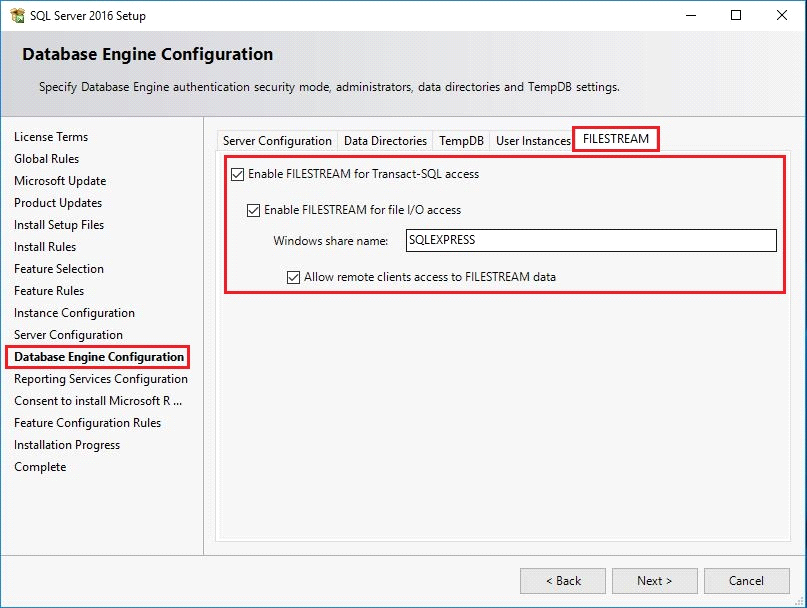
Sql Server 2016 Express

* Download and install Sql Server 2016 Express
* Use the default selections for every part of the installation except the following:
* *Server Configuration* - Ensure "Mixed Mode" is selected for authentication

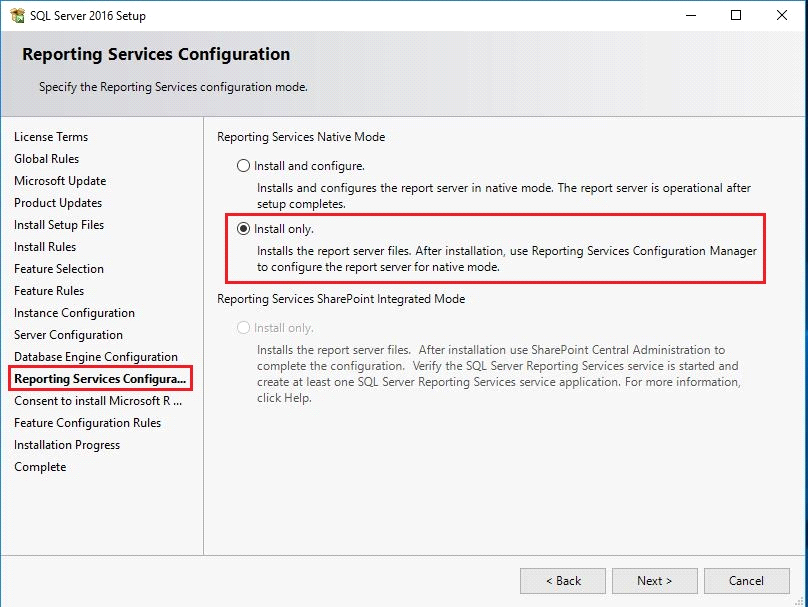


(assign "r2g2" as the password for the ***sa*** user)

* *FILESTREAM* - Ensure all FILESTREAM options are enabled



* *Reporting Services Configuration* - No need to configure SSRS

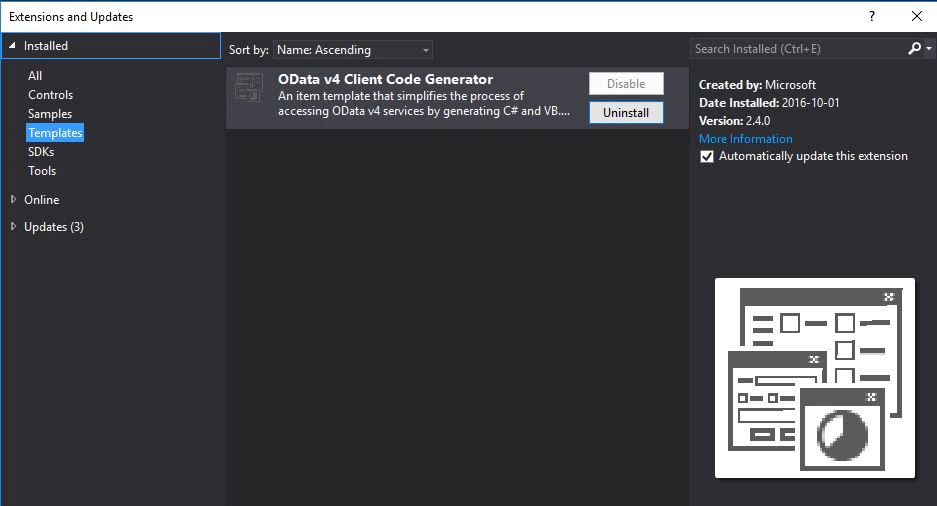


Phidget Drivers

* Download and install the **Phidget21** drivers from <http://www.phidgets.com/docs/OS_-_Windows#Quick_Downloads>

Visual Studio

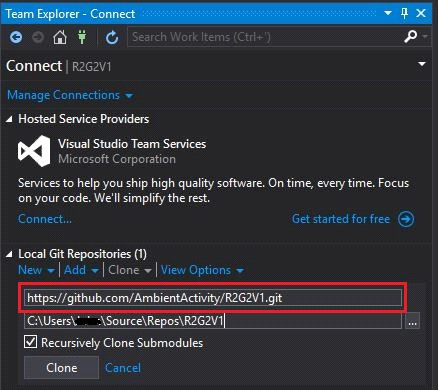
* Download and install **Visual Studio**
* As of this writing the version being used is **2015 Community** which can be downloaded from <https://www.visualstudio.com/downloads/>
* Open **Visual Studio** (as administrator)
* In **Visual Studio** goto Tools -> Extensions and Updates...
* Install the Template **OData v4 Client Code Generator**



Clone the Repository

* In **Visual Studio** Go to *Team Explorer* and clone the repository at

<https://github.com/AmbientActivity/R2G2V1.git>



Install Needed Visual Studio Updates

Upon first opening the solution in **KeebeeAAT.sln** in Visual Studio you will be prompted for required updates.

1) For the *Video Capture Service* project to build the following SDK is required:



2) For the *Bluetooth Beacon Watcher Service* project to build the following SDK is required:



Allow Visual Studio to download and install the necessary updates. Note that they are large and that this process will take quite a while to complete (sorry).

If not automatic, download and install the **Windows Phone 8.0 SDK** from:

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

If not automatic, download and install the **Windows Tools for Universal Windows Apps** from: <https://www.visualstudio.com/vs/universal-windows-platform/>

Build Solution

* Open the solution in **KeebeeAAT.sln** in Visual Studiofrom the folder

*C:\Users\<Username>\Source\Repos\R2G2V1*

**Before Building**:

* The NReco video conversion dll’s need to be copied to their correct locations
* Navigate to the folder *C:\Users\<Username>\Source\Repos\R2G2V1\3rdParty*, right-click on the file **Install3rdParty.ps1** and select "Run with PowerShell"
* This will copy the NReco dll's to the bin folders of the projects **Keebee.AAT.ThumbnailGeneration** and **Keebee.AAT.VideoConversion** (both under the "Shared" folder of the repository)

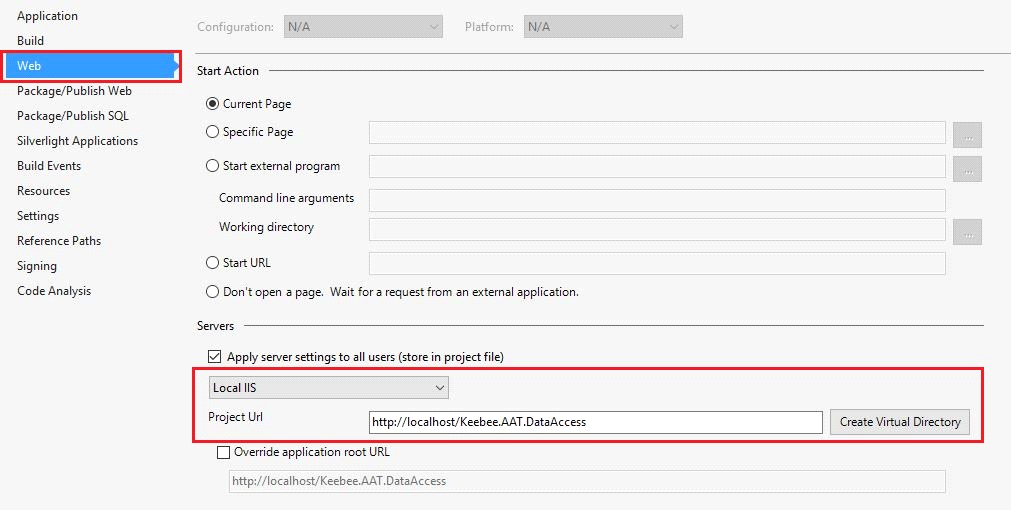
**Build:**

* Build the solution in *Debug Mode*
* (Nuget packages will download themselves before the build begins)
* Build the solution in *Release Mode*

Create Virtual Directories

* In Visual Studio find the project **Keebee.AAT.DataAccess** in the solution (under the folder "Data") and select "Properties"
* Create the virtual directory as shown below

*Note: Need to launch Visual Studio as administrator for this to succeed*



* Repeat above steps for the project **Keebee.AAT.Operations** (under the folder "API" in the solution)
* Repeat above steps for the project **Keebee.AAT.Administrator** (under the folder "WebUI" in the solution)

Flash Apps

* All SWF files are prebuilt, they simply need to be copied to the correct location
* The built SWF's reside in the folder *C:\Users\J<Username>\Source\Repos\R2G2V1\Flash\Builds*
* Their destination will be in the bin *Debug* and *Release* folders of **Keebee.AAT.DIsplay**
* To copy them, navigate to the folder *C:\Users\<Username>\Source\Repos\R2G2V1\DebugScripts* and run the batch file **CopyFlashBuildsLocal.bat**
* At this point, there is nothing more that needs to be done regarding the flash applications, however if there is a need to build them from scratch see the information below on how to download/install/configure *FlashDevelop*

FlashDevelop

* In the event that you wish to build the SWF's from scratch
* Download and install the free **FlashDevelop** IDE for compiling ActionScript source code into executable SWF’s. It can be downloaded from <http://www.flashdevelop.org/>
* Download and install the SDK *Flex 4.6.0, AIR 23.0* (or whatever the latest version is) and configure FlashDevelop to use it through the *Project->Properties* menu (SDK tab)
* Download the latest version of *Flash Player Debug* and configure FlashDeveloper to use it through the *Tools->Program Settings* menu. Look for *FlashViewer* in the left panel and set the debugger path for the property *External Player Path*

Build Flash Apps

* Open **FlashDevelop**
* Open and build the project **SlideViewer.as3proj** from the folder:

*C:\<Username>\John\Source\Repos\R2G2V1\Flash\SlideViewer*

* Open and build the project **MatchingGame.as3proj** from the folder:

*C:\Users\<Username>\Source\Repos\R2G2V1\Flash\MatchingGame*

* Open and build the project **PaintingActivity.as3proj** from the folder:

*C:\Users\<Username>\Source\Repos\R2G2V1\Flash\PaintingActivity*

* Open and build the project **BalloonPoppingGame.as3proj** from the folder:

*C:\Users\<Username>\Source\Repos\R2G2V1\Flash\BalloonPoppingGame*

Deploy R2G2

* Navigate to the folder: *C:\Users\<Username>\Source\Repos\R2G2V1\Deployment*
* Run the file **DeployAll.bat** as administrator

Note: If the output of the script displays "Building Debug..." for more than 20 -30

seconds simply close the command window and re-launch the script.

(This is a known issue that has yet to be resolved. It generally only happens the first time a build is attempted)

Install R2G2

* After **DeployAll.bat** batch script has succeeded (above) navigate to the folder:

*C:\Deployments\Install\*

* Right-click and execute the batch file **INSTALL\_R2G2.bat** as administrator
* It will install everything needed to run/configure/develop *R2G2* on your system

Note 1:This will overwrite the locations of the web applications defined in IIS that were setup in the *Create Virtual Directories* step from above and they will now be pointed to their respective locations in *C:\Deployments\Web*

When the solution is rebuilt in Visual Studio however, it will update the IIS paths to use the development locations from the repository again (as in the *Create Virtual Directories*  step from above)

On the development computer, all web applications should be running from their locations in the repository

Note 2: In production it is necessary to set the *DefaultAppPool* identity to **webuser**

(which will be created by the script **4\_CreateLocalWebuser.bat**) but it is not

necessary when running the Administrator web application locally.

In production, it is also necessary to set the *Pass-through authentication* for

the Administrator web application to **webuser** but it is not necessary when

running the Administrator web application locally.

The webuser still needs to be created however because *Sql Server* requires it

Installing R2G2 in Stages

* An alternative to running the above **INSTALL\_R2G2.bat** is torun the installation scripts in individual stages
* This is useful for troubleshooting in case any part of the installation failed
* If any part of the installation failed, navigate to *C:\Deployments\Install*, right-click on **UNINSTALL\_R2G2.bat** and run as administrator
* This will completely uninstall any components that installed successfully and allow you to start from a clean slate
* To install R2G2 in stages, navigate to the folder *C:\Deployments\Install\Utility\\_Install*, right-click on each of the numbered batch files (starting with **1\_CreateEventLogSources.bat**) and run them sequentially (again as administrator) and determine which one of the steps is failing
* The error message will offer some clue as to where the problem is occurring and hopefully give you enough information to allow you to figure it out
* Most likely causes are:
  + A Windows component is missing
  + SQL Server Express was not installed correctly

Post Installation

* The installation created several new *Scheduled Tasks* on your system that should all be disabled at this point for development purposes
* Go to *Computer Management -> Task Scheduler -> Task Scheduler Library* and disable the following tasks:
  + R2G2 - Backup
  + R2G2 - Display Launch
  + R2G2 - Event Log Export
  + R2G2 – System Restart
  + R2G2 Video Capture File Cleanup

Run R2G2 Administrator Web Application

* Open a browser and navigate to <http://localhost/Keebee.AAT.Administrator/>
* Select "Administrator" as the *Username*, enter "@dmin" for the password and login
* Navigate through all the screens to ensure IIS is healthy and all the endpoints are being reached
* In the *Public Library* menu (under *Manage*) ensure there is at least one media file seeded for all media types
* Logout

Run R2G2 Simulator and Display App in Debug Mode

* Navigate to the folder *C:\Deployments\UI\Simulator\1.0.0.0*
* Make a shortcut to **Keebee.AAT.Simulator.exe** and place it on your desktop

- Name it "Simulator"

* Navigate to the folder *C:\Deployments\UI\Display\1.0.0.0\Debug*
* Make a shortcut to **Keebee.AAT.Display.exe** and place it on your desktop

- Name it "DEBUG Display"

* Navigate to the folder *C:\Deployments\UI\Display\1.0.0.0\Release*
* Make a shortcut to **Keebee.AAT.Display.exe** and place it on your desktop

- Name it "RELEASE Display"

* Double-click the shortcut "Simulator" from the desktop

**Note:** Once it opens, move the window to the right-hand side of your screen

* Double-click the shortcut "DEBUG Display" from the desktop
* It will appear in the top left corner of your screen and loop through a 3 second ambient video
* Click all the "Response" buttons of the **Simulator** to ensure all activities respond
* To exit the **Display App**, click the button "Kill Display" on the Simulator (and close the Simulator)

Phidgets

* If you have a Phidget Interface kit installed, these can be tested now
* They should be able to trigger all response types from the *Display App*
* For this functionality, you need to run the *Administrator Web Application*and navigate to *System -> Configuration -> Phidgets* to examine the default *Phidget Configuration*
* Here you have 2 options:
  + - * 1. a) Add a new configuration and assign phidgets per how your phidgets are physically plugged in

b) Reconfigure your phidget sensors and inputs to correspond to the *Default Configuration*

* In the folder *C:\Users\<Username>\Source\Repos\R2G2V1\DebugScripts* there is a batch file
* called **SeedConfigurationJohn.bat**which I use to add a configuration to the system that matches my phidget setup
* Multiple *Phidget Configurations* can be defined but only one can be activated at any given time

Run R2G2 Display App in Release Mode

* Navigate to *C:\Deployments\UI\Display\1.0.0.0\Release*
* Double-click the shortcut "RELEASE Display" on the desktop
* Test the application with real phidget sensor activity
* To exit hit *WindowsKey-D* which allows access to the desktop, double-click the "Simulator" shortcut and click "Kill Display"

Beacons

* R2G2 uses *Beacon Advertisements* to determine the identity of a specific resident who wishes to engage with the unit (which implies they have personalized media defined in the database)
* The *Bluetooth Beacon Watcher Service* performs the task of handling these advertisements
* Specifically, it listens for beacons that broadcast a message using the *iBeacon* format
* The Company Name is defined in the *CompanyUUID* portion of the advertisement frame
* The *FacilityId* is defined in the in the "major" portion of the data frame

**Note:** *FacilityId is not implemented at this time and is currently set to "0"*

* The *ResidentId* is defined in the "minor" portion of the data frame

The **App.config** file under the *Bluetooth Beacon Watcher Service* contains App Settings that define the constant values for *CompanyUUI* and *FacilityId*

*CompanyUUID* appears as a 16-byte hexadecimal string:

00-00-00-00-00-00-00-00-00-00-00-00-00-41-41-54 = "AAT"

*FacilityId* appears as the integer "0"

The *Beacon Service* strips off the *CompanyUUID* from the frame to confirm its validity.

It then reads the "minor" portion of the frame to obtain the *ResidentId* (an integer).

This value is then sent via a Message Queue to the *State Machine Service* which then proceeds to load that resident's profile into memory

Beacon Monitor

* For troubleshooting purposes, there is a *Beacon Monitor*project (under the *folder C:\Deployments\UI*)
* It is included in the deployment (unlike the *Phidget Monitor* application which is discussed below)

**Note:** The *Beacon Monitor* will only function correctly if it is launched after:

1) The *Bluetooth Beacon Watcher Service* has been installed (Debug or Release mode)

2) The *Display App* is running (Debug or Release mode)

Phidget Monitor

* For troubleshooting purposes, there is a *Phidget Monitor*project in the repository under *UI*
* This is to test that the phidgets are communicating with the *Phidget Service* correctly
* Due to its limited use, it does not get deployed but can be run from its location in the repository
* In order to use this helper application 2 new message queues are required
* To create them run the script *C:\Deployments\Install\Utility\CreateMessageQueuesDev.bat* (as admin)

**Note:** The *Phidget Monitor* will only function correctly if it is launched after:

1) The *Phidget Service* has been installed in Debug mode

2) The *Display App* is running (in Debug or Release mode)

Windows Services (Debug Mode)

* When debugging any of the Services it is recommended to install them using the script:

***C:\Users\<Username>\Source\Repos\R2G2V1\DebugScripts\TEST - Install Services .bat***

* To uninstall them, use the script:

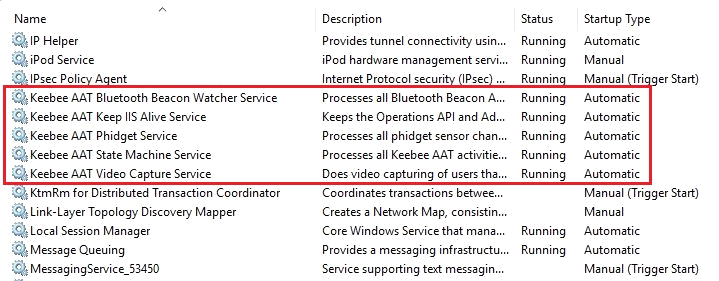
***C:\Users\<Username>\Source\Repos\R2G2V1\DebugScripts\TEST - Install Services .bat***

* Make a desktop shortcut to these files and set their properties so that they will always execute as administrator

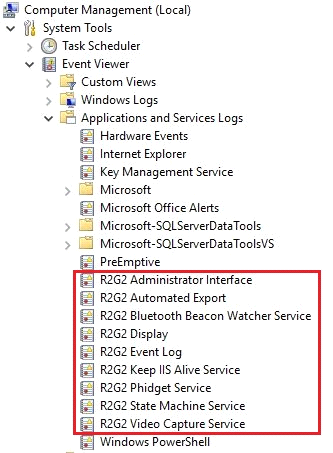
-------------------------------------------------------------------------------------------------------------------

Troubleshoot / Verify Installation

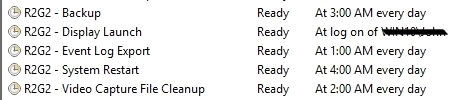
*Computer Management* - Verify that all **Keebee AAT Services** are running



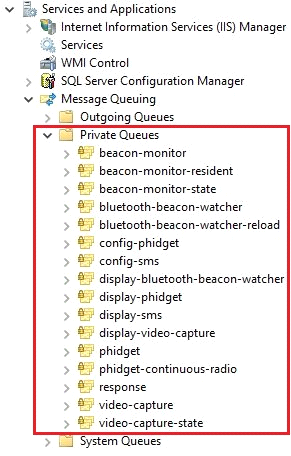
*Computer Management* - Verify that all **Event Log Sources** were created



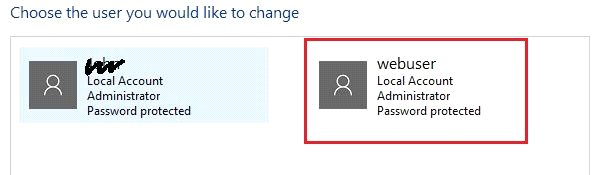
*Computer Management* - Verify that all **Scheduled Tasks** were created



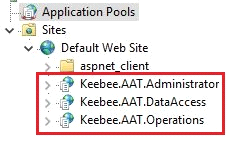
*Computer Management* - Verify that all **Message Queues** where created



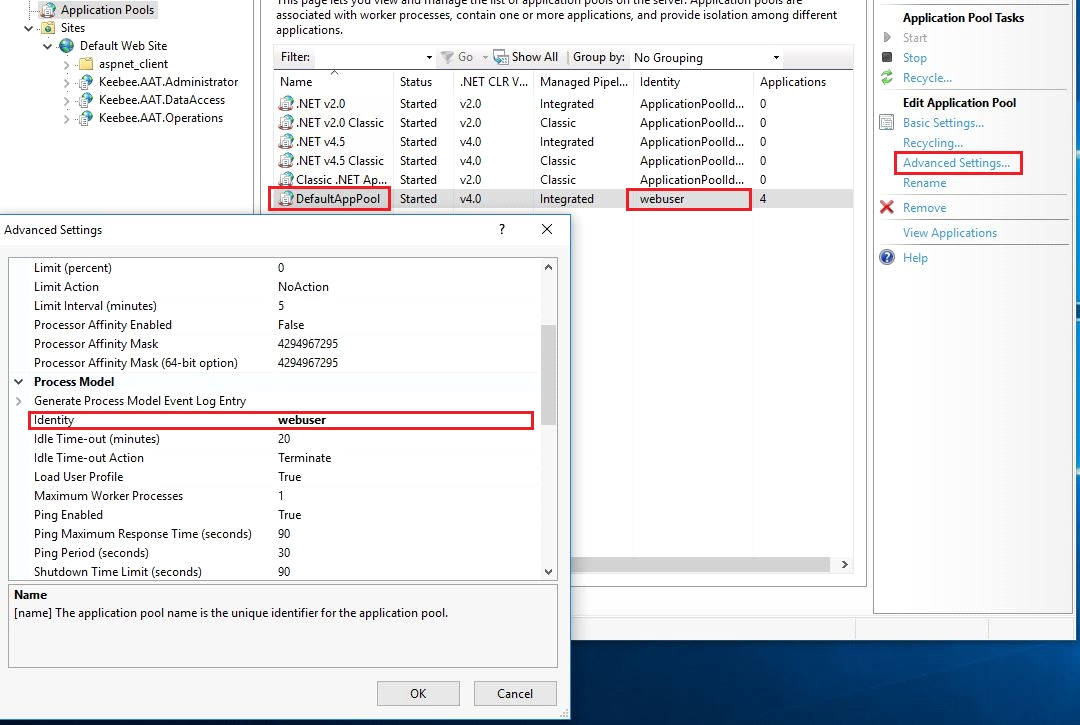
*User Accounts* - Verify that a new **webuser** was created



*IIS Manager ­-* Verify that all **Web Applications** were created



*IIS Manager ­- V*erify that the *DefaultAppPool* uses **webuser** for its identity



Verify InvokeSqlQuery Module Registration

* Navigate to "C:\Windows\System32\WindowsPowerShell\v1.0\Modules"
* Ensure that the folder "InvokeSqlQuery" was created
* Ensure the files *InvokeSqlQuery.psd1* and *InvokeSqlQuery.psm1* exist inside
* ***Note: These modules are used for initial database creation and seeding only***

