Telemetry System Testing Environment & Testing Scripts

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Ver.** | **Description** | **Author** |
| March 13, 2010 | 1.0 | Initial Composition | Daniel Johnson |
| May 2, 2010 | 1.1 | Updated Testing Environment | Daniel Johnson |
| May 2, 2010 | 1.2 | QA Revision | John Schmidt |

Table of Contents

[1. Introduction 4](#_Toc257889433)

[1.1 Purpose 4](#_Toc257889434)

[2. Testing Environment 4](#_Toc257889435)

[3. Testing Scripts 8](#_Toc257889436)

[3.1 Testing Script: Display Device Telemetry Data 8](#_Toc257889437)

[3.1.1 Test Setup 8](#_Toc257889438)

[3.1.2 Test Steps: Happy Day 9](#_Toc257889439)

[3.1.3 Happy Day Test Outcomes 9](#_Toc257889440)

[3.1.4 Test Steps: Rainy Day 9](#_Toc257889441)

[3.1.5 Rainy Day Test Outcomes 9](#_Toc257889442)

[3.2 Testing Script: Parse Telemetry Data 9](#_Toc257889443)

[3.2.1 Test Setup 9](#_Toc257889444)

[3.2.2 Test Steps: Happy Day 9](#_Toc257889445)

[3.2.3 Happy Day Test Outcomes 9](#_Toc257889446)

[3.2.4 Test Steps: Rainy Day 10](#_Toc257889447)

[3.2.5 Rainy Day Test Outcomes 10](#_Toc257889448)

# Introduction

## Purpose

This document is intended to give outside developers the basic testing environment and tests that can be run on the web system portion of our project. The document will first cover the testing environment then move onto testing scripts.

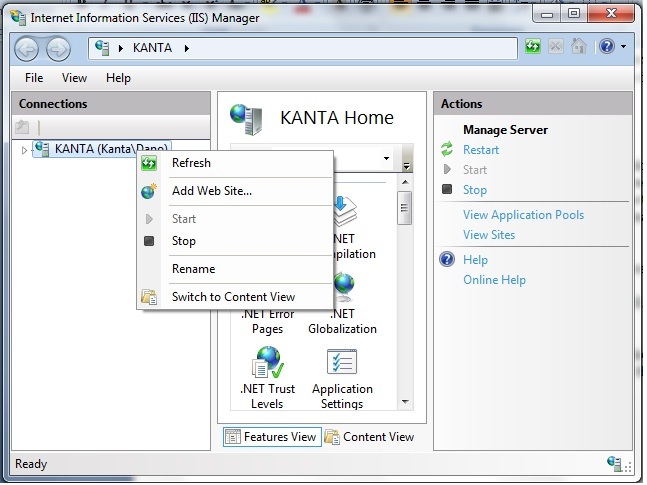
# Testing Environment

The testing environment is recommended to be done on Windows 7, but earlier versions may work with some compatibility issues that will have to be addressed by the tester. Setup examples will be demonstrated on a Windows 7 machine. The testing environment will require the tester to have these components installed on the testing machine.

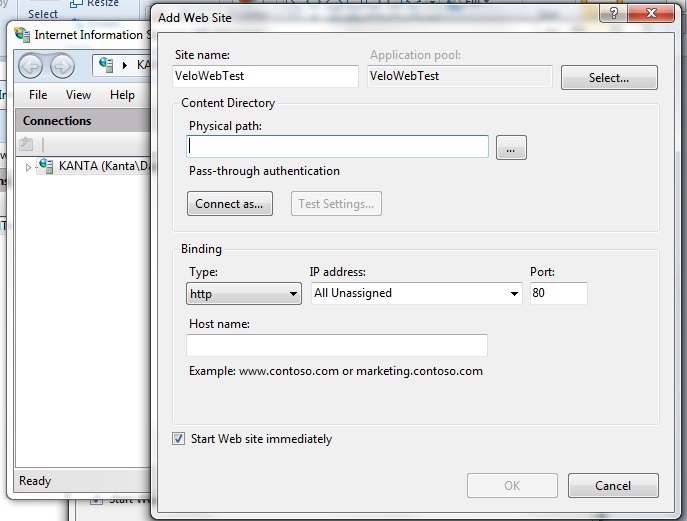
1. Microsoft SQL Server 2008
2. Internet Information Services 7 (IIS7)

Once the testing machine has the necessary software, they will need to change some configurations for IIS7 to function properly.

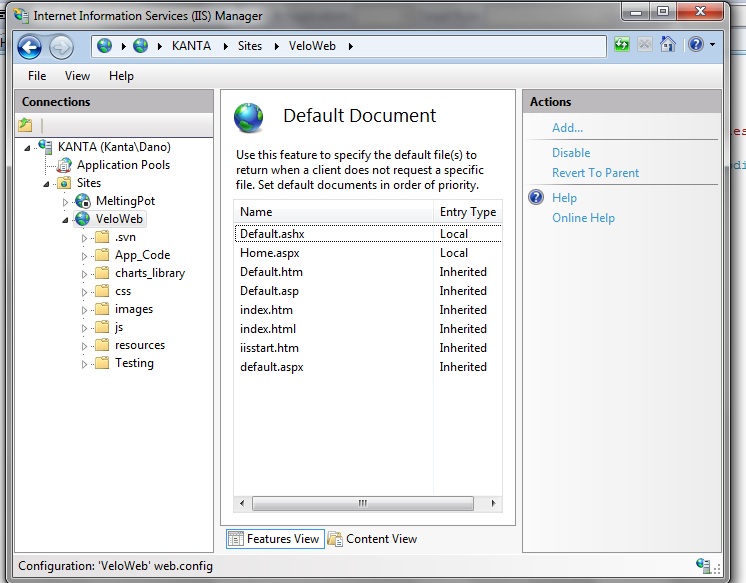
1. First you want to change the security permissions on the folder that will be containing the web system. IIS\_IUSRS will have to be added with full control of that folder. This is the authentication IIS7 will be using.
2. No you should add our web system to IIS 7, to do this you will start IIS7. Once IIS7 is loaded, right-click on the testing machines name in the list found at the left of the window, and select add website.



1. You should now see this window, for the server name enter “VeloWebTest” or any identifier you would like. In the physical path you will need to input the directory “\Telemetry Server\Telemetry Frontend”. It will be in our provided deliverable. For simplicity I recommend you leave binding to its default settings. If you do you will be able to navigate to the site once setup is complete by entering <http://localhost/> in your browser. Make sure to change Application Pool to DefaultAppPool as well. This is important so that IIS7 can authenticate with the database and access it correctly.

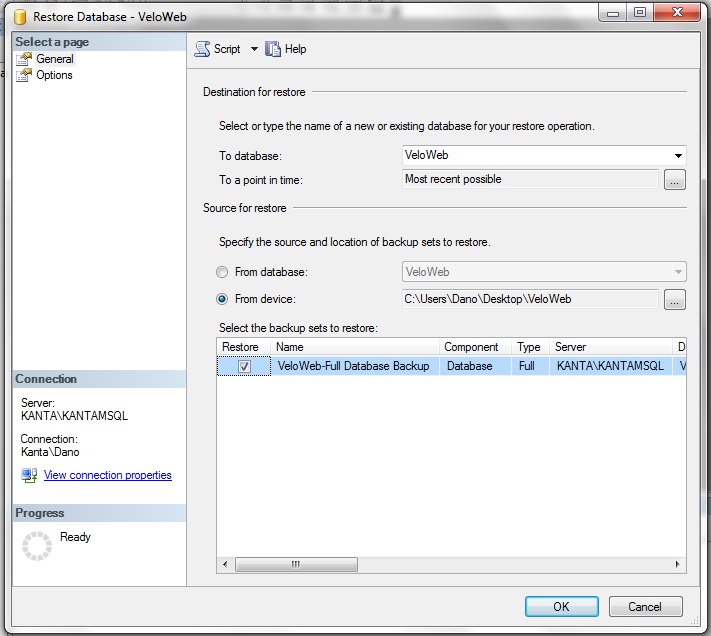


1. Now that the website is created, the default file needs to be changed. Select the website you just created from the list on the left side of the window again. Then double click on the Default File icon in the IIS setup area. Once you are to the page shown in the following image, right click in the list area below default.aspx and select Add… then type in Home.aspx and move it to the top. Your default document page should closely resemble the one in the following image now.

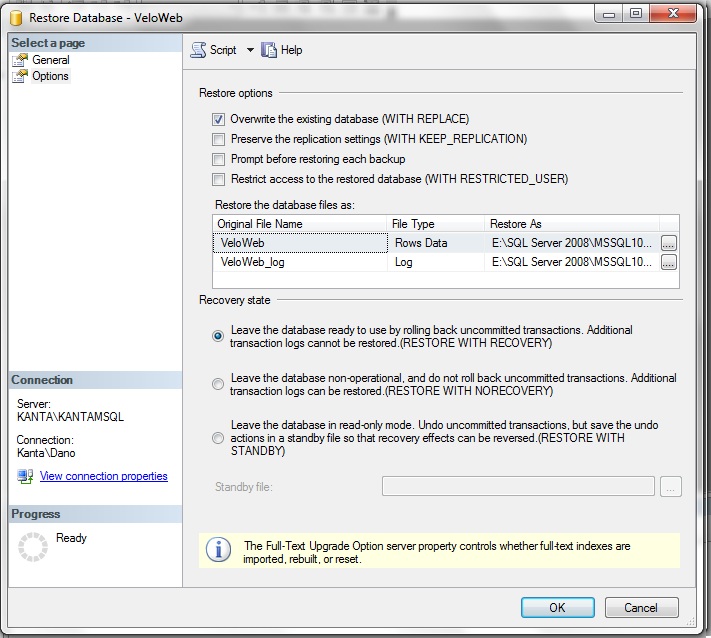


Now that IIS 7 is installed, the web system should be ready for testing as soon as you recover the database from the recovery file provided. To recover the database please follows the steps provided below.

1. Start Microsoft SQL Server Management Studio 2008 and connect to the Microsoft Server SQL Server database engine.
2. Create a database and name it “TricsWeb”
3. Now navigate to the Install.aspx page through your web browser, this will finish setting up the database tables for you and modify the web.config.
4. Once that is done, right-click on the database and select Tasks->Restore->Database…
5. Once there you will need to remove all sources from the list near the bottom of the window, then locate and add the backup file Testing Documentation\VeloWebBack.bak, your screen should now look like the image below.



1. Before you hit accept, there is a setting you need to change on the Options page.
2. Once on the options page, check-mark “Overwrite with existing database (WITH REPLACE)” it should look like this.



1. Once the above steps are done hit ok.
2. Now you will need to open the Security->Logins tab for your database server and add the IIS app pool as a database user. What you will be looking for is IIS APPPOOL\DefaultAppPool.
3. Once the app pool exists as a database login, you need to add it to the trics web database specifically. To do so navigate to Databases->TricsWeb->Security->Users and add the default app pool as a user. Name it DefaultAppPool and tie it to the IIS APPPOOL\DefaultAppPool login.
4. Once all of these steps are completed you are ready to test.

# Testing Scripts

This section will cover all testing scripts for testing our use cases in the alpha release. Testers are welcome to look at the associated use cases for each test script to better understand what they are testing.

## Testing Script: Display Device Telemetry Data

This test script ties to the use case Display Device Telemetry Data in “\Project Artifacts\Use Cases\Web System” and covers the test setup, steps and outcomes. This test will display telemetry data from the Velomobile and should only have a Light sensor graph with one piece of data in it. Once verified the Devices.aspx page will be used to test other devices.

### Test Setup

To begin this test, open your browser then navigate to <http://localhost/Devices.aspx>.

### Test Steps: Happy Day

These steps will walk the tester through a happy day scenario for this use case. In this test we are verifying that the device drop down list is populated and that we can select a device to view telemetry data.

1. Using the Change Device drop down list select Velomobile.
2. Wait for the flash content to finish loading.

### Happy Day Test Outcomes

There should now be a light sensor graph loaded with 2 pieces of data in it, the data should be approximately 500.

### Test Steps: Rainy Day

These steps will walk the tester through a rainy day scenario for this use case. In this test the Rainy Day device has no sensors and the error will be handled.

1. Using the Change Device drop down list select Rainy Day.
2. Wait for the page to finish loading.

### Rainy Day Test Outcomes

There should now be a text message in place of where the graphs would be, explaining that there are no sensors associated to the Rainy Day device.

## Testing Script: Parse Telemetry Data

This test script ties to the use case Display Device Telemetry Data in “\Project Artifacts\Use Cases\Web System” and covers the test setup, steps and outcomes.

### Test Setup

To begin this test, open browser 1 then navigate to <http://localhost/Devices.aspx>

You will also need to open a second browser (browser 2) and navigate to <http://localhost/Protocol.Live.aspx>

Now position both browsers so you have a view of both simultaneously.

In browser 1 select Velomobile from the Change Device drop down list.

### Test Steps: Happy Day

These steps will walk the tester through a happy day scenario for this use case. In this test we are adding data to the database using the Live Parser.

1. In browser 2 click on the parse button
2. Wait for the response text box to fill with a response.
3. Watch browser 1 for an update to the Light sensor graph, there should be a new value of 750 appended to the graph.
4. Repeat steps as necessary, changing the request text box XML value for light sensor before proceeding to step 1 to observe different values being parsed.

### Happy Day Test Outcomes

Depending on the number of tests, you should now see new data in the light sensor graph

### Test Steps: Rainy Day

These steps will walk the tester through a rainy day scenario for this use case. The rainy day situation for this use case is that there is no bump sensor for the Velomobile, thus there will be no change on the system.

1. In browser 2 change the request text box to:

<Velomobile> <bump>123 </bump></Velomobile>

1. In browser 2 click on the parse button
2. Wait for the response text box to fill with a response.
3. Watch browser 1 for an update for roughly 5 seconds.

### Rainy Day Test Outcomes

There should be no change to browser 1 as there is no bump sensor registered in the database for the Velomobile.