



**Installing the ASP.NET VETtrak API  
onto IIS 7 or Later**

<b>Installing the VETtrak APIs onto IIS 7 or Later</b>	<b>3</b>
1 Step 1- Install/Check IIS 7, 7.5, 8 or 8.5.....	3
Windows Vista or Windows 7 .....	3
Windows 8 or 8.1 .....	4
Windows Server 2008 .....	5
Windows Server 2008 R2 .....	6
Windows Server 2012 or 2012 R2 .....	7
2 Step 2 - Set Up and Configure VETtrak ASP.NET API.....	8
3 Step 3 - Adding an Application Pool.....	9
4 Step 4 - Add Application.....	11
5 Step 5 - Test that the VETtrak API is Accessible.....	13
6 Step 6 - Test the VETtrak Enrol API with the Test Harness.....	16

# 1 Installing the VETtrak APIs onto IIS 7 or Later

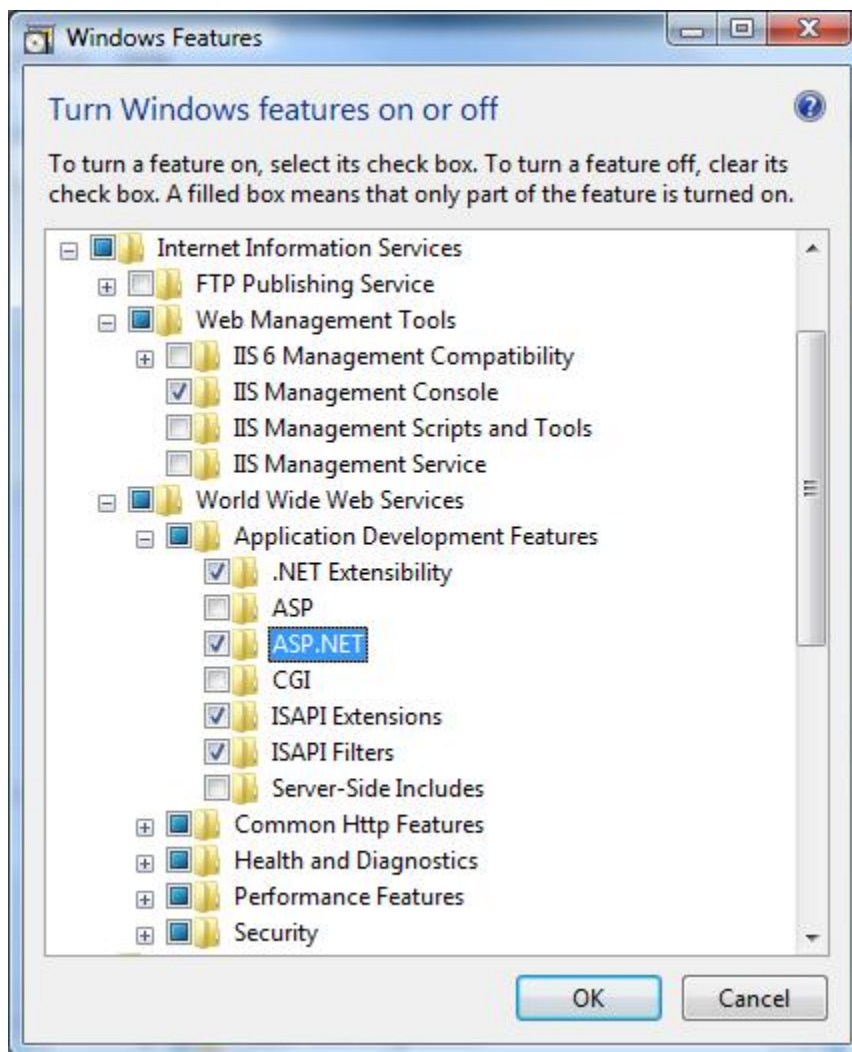
This document describes how to install the ASP.NET VETtrak APIs onto a Microsoft Internet Information Services 7 (on Windows Vista or Server 2008), 7.5 (on Windows 7 or Server 2008 R2), 8 or 8.1 (on Windows 8 or Server 2012) server, and how to configure it to run.

## 1.1 Step 1- Install/Check IIS 7, 7.5, 8 or 8.5

Ensure that the .NET Framework version 3.5, .NET Framework version 2.0 with at least Service Pack 2, and Internet Information Services (IIS) 7 or later is installed on the server machine. At a minimum, the IIS Management Console and ASP.NET components (which will automatically include its dependencies such as .NET Extensibility, ISAPI Extensions and ISAPI Filters) must be installed. Additional IIS features may be installed if desired, but are not required by the VETtrak APIs.

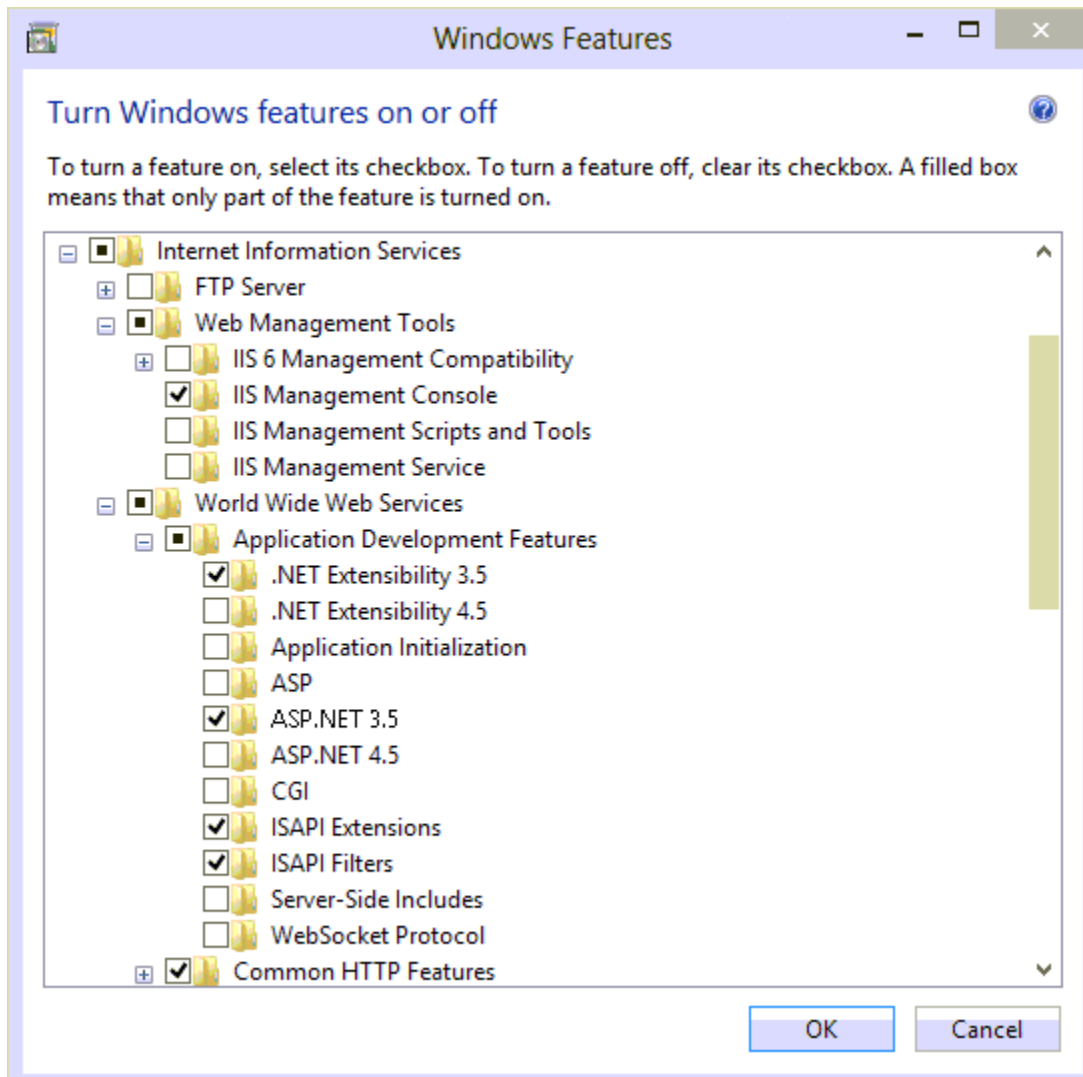
### 1.1.1 Windows Vista or Windows 7

In Windows Vista / 7, IIS can be installed or the installation checked by going to **Start > Control Panel > Programs and Features > Turn Windows Features on or off**.



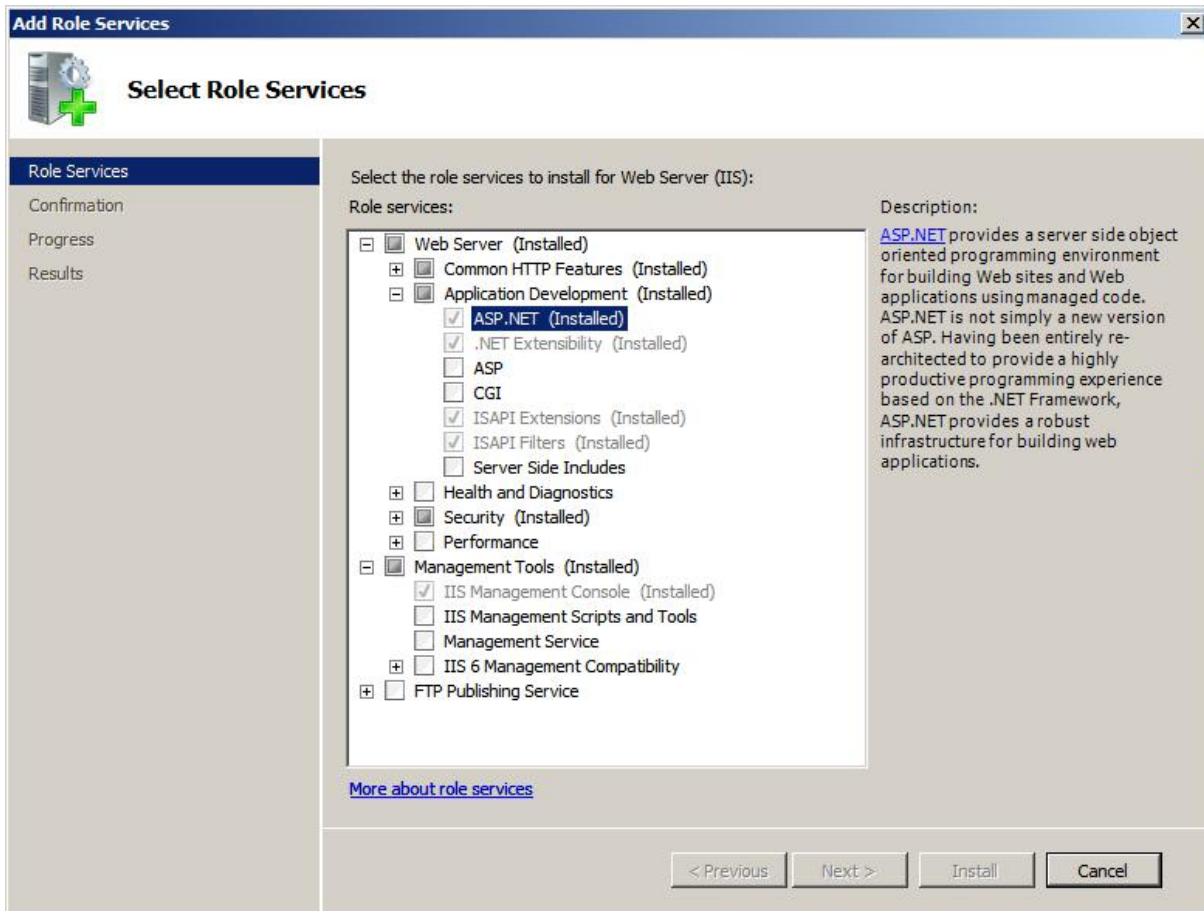
### 1.1.2 Windows 8 or 8.1

In Windows 8 or 8.1, IIS can be installed or the installation checked by opening the Control Panel, then opening **Programs and Features** and clicking **Turn Windows features on or off**.



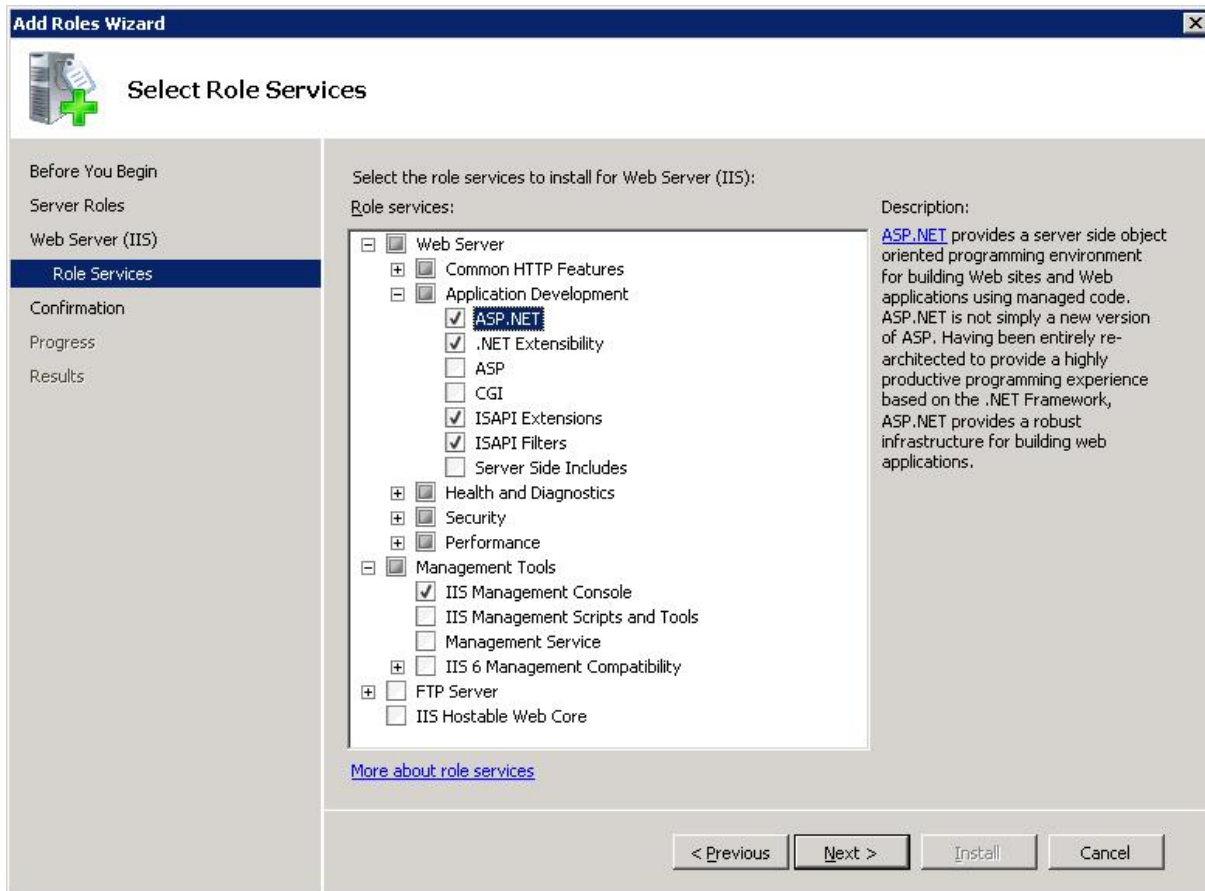
### 1.1.3 Windows Server 2008

In Windows Server 2008, IIS can be installed or the installation checked by going to **Start > Control Panel > Administrative Tools > Server Manager > Roles**, and adding or editing the **Web Server (IIS)** role:



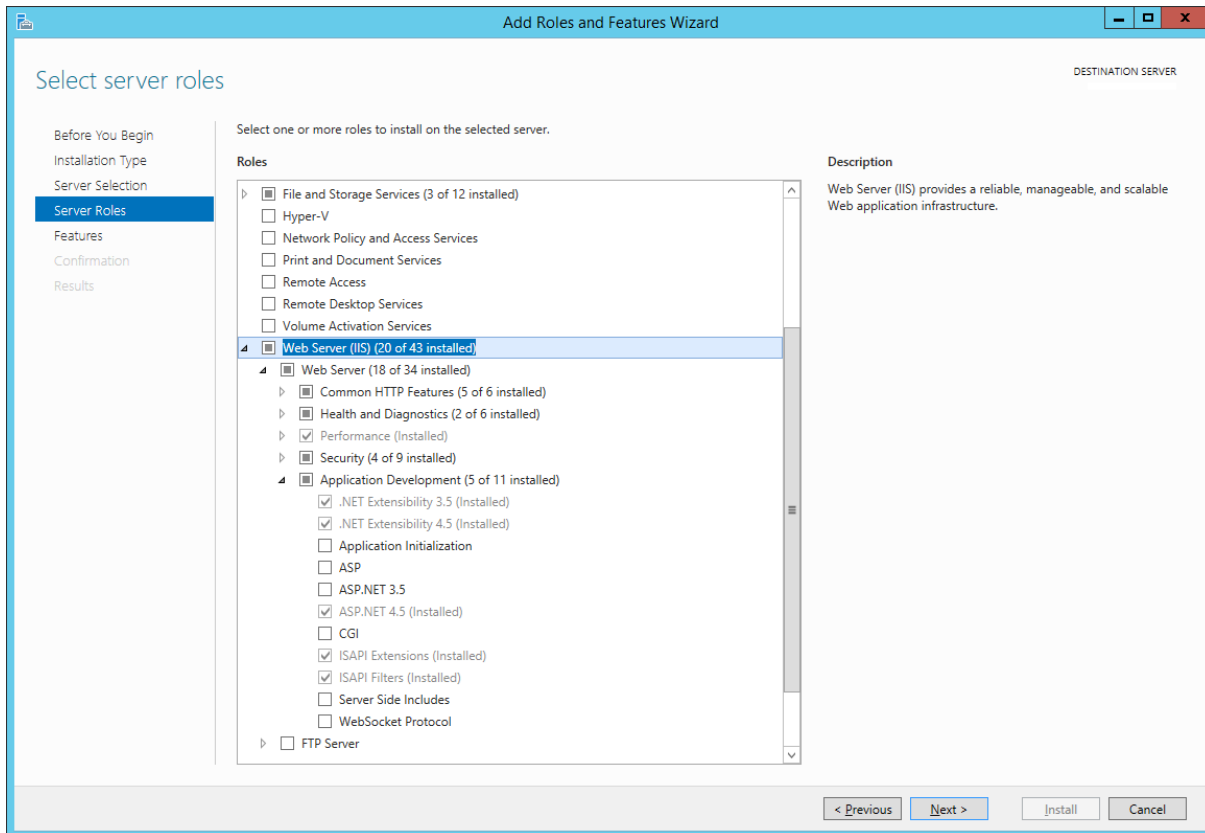
### 1.1.4 Windows Server 2008 R2

In Windows Server 2008 R2, IIS can be installed or the installation checked by going to **Start > Control Panel > Administrative Tools > Server Manager > Roles**, and adding or editing the Web Server (IIS) role:



### 1.1.5 Windows Server 2012 or 2012 R2

In Windows Server 2012 or 2012 R2, IIS can be installed by opening the Server Manager, going to the “Manage” menu, selecting “Add Roles and Features”, and adding the Web Server (IIS) role. For details, refer to <http://www.iis.net/learn/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012>.



## 1.2 Step 2 - Set Up and Configure VETtrak ASP.NET API

In order for the API to run, a valid registration key for the VETtrak API must be entered into the VETtrak database using the VETtrak software, in **File > Preferences > Registration Keys**.

Obtain the VETtrak ASP.NET API ZIP file from VETtrak support. This can be downloaded from the Downloads page in the Support Centre.

Before unzipping the ZIP file, right-click it and go to **Properties**. If there is an **Unblock** button on the **General** tab, saying that the file has come from the internet, click the **Unblock** button followed by **OK**.

Extract the contents of the VETtrakAPI folder from within the VETtrak ASP.NET API ZIP file to a new folder on the web server. It may be placed in a folder in the web root if desired, which by default is C:\inetpub\wwwroot.

Configure the API by editing the web.config file. This is an XML file that can be opened in a text editor. Find the `<connectionStrings>` element. Within this is an `<add>` element containing details of how to connect to the database. The `<add>` element must contain the following three attributes:

- The `name` attribute must be "VETtrakDatabase".
- The `providerName` attribute must be set to "FirebirdSql.Data.FirebirdClient" if using a Firebird database, or "System.Data.SqlClient" if using a SQL Server database.
- The `connectionString` attribute is a standard connection string that defines the database server, database filename (for Firebird) or database name (for SQL server), and the database username and password. Examples follow.

The contents of the `connectionString` should be the same as VETtrak uses to connect to the database – view these in VETtrak by going to **File > Preferences > Database connection**.

Note that, once the VETtrak API application is set up in IIS, the connection string can be easily changed using the Connection Strings icon in the application in IIS Manager – however, the `providerName` can only be changed by editing the web.config file directly.

### Firebird

If using a Firebird database, the add element and connection string should be in the format:

```
<add name="VETtrakDatabase" providerName="FirebirdSql.Data.FirebirdClient"
connectionString="User=<databaseUser>; Password=<databasePassword>;
Database=<pathAndFilenameOfDatabaseFileOnServer>;
DataSource=<serverNameOrIP>;Dialect=3;" />
```

For example:

```
<add name="VETtrakDatabase" providerName="FirebirdSql.Data.FirebirdClient"
connectionString="User=SYSDBA; Password=masterkey; Database=C:\Program Files
\VETtrak\Data\VETtrak.gdb; DataSource=192.168.0.40; Dialect=3;" />
```

The components of the `connectionString` are:

- **User** – the Firebird username to connect to the database (not the username used to log into VETtrak). This is almost always "SYSDBA"
- **Password** – the Firebird password to connect to the database (not the password used to log into VETtrak). This is almost always "masterkey"
- **Database** – the full file path on the server to the database file (usually called VETtrak.gdb)
- **Datasource** – the IP address or resolvable name of the server that is running Firebird and on which the database file is located
- **Dialect** – leave this as "3"

### SQL Server with SQL server authentication

If using a SQL Server database, with SQL server authentication, the add element and connection string should be in the format:

```
<add name="VETtrakDatabase" providerName="System.Data.SqlClient"
```



```
connectionString="Data Source=<serverNameOrIP>\<instanceName>;Initial
Catalog=<databaseName>;User Id=<databaseUser>;
Password=<databasePassword>;"/>
```

For example:

```
<add name="VETtrakDatabase" providerName="System.Data.SqlClient"
connectionString="Data Source=192.168.0.40\SQLEXPRESS;Initial
Catalog=VETtrak;User Id=sa;Password=sa;"/>
```

The components of the `connectionString` are:

- `Datasource` – the IP address or resolvable name of the server that is running SQL Server, followed by a backslash and then the name of the SQL Server instance that contains the VETtrak database
- `Initial Catalog` – the name of the VETtrak database on that server and instance
- `User Id` – the SQL Server username to connect to the database (not the username used to log into VETtrak)
- `Password` – the SQL Server password to connect to the database (not the password used to log into VETtrak)

## SQL Server with Windows authentication

If using a SQL Server database, with Windows authentication, the add element and connection string should be in the format:

```
<add name="VETtrakDatabase" providerName="System.Data.SqlClient"
connectionString="Data Source=<serverNameOrIP>\<instanceName>;Initial
Catalog=<databaseName>;Integrated Security=true"/>
```

For example:

```
<add name="VETtrakDatabase" providerName="System.Data.SqlClient"
connectionString="Data Source=192.168.0.40\SQLEXPRESS;Initial
Catalog=VETtrak;Integrated Security=true"/>
```

The components of the `connectionString` are:

- `Datasource` – the IP address or resolvable name of the server that is running SQL Server, followed by a backslash and then the name of the SQL Server instance that contains the VETtrak database
- `Initial Catalog` – the name of the VETtrak database on that server and instance
- `Integrated Security` – leave this as “true” to use Windows Authentication

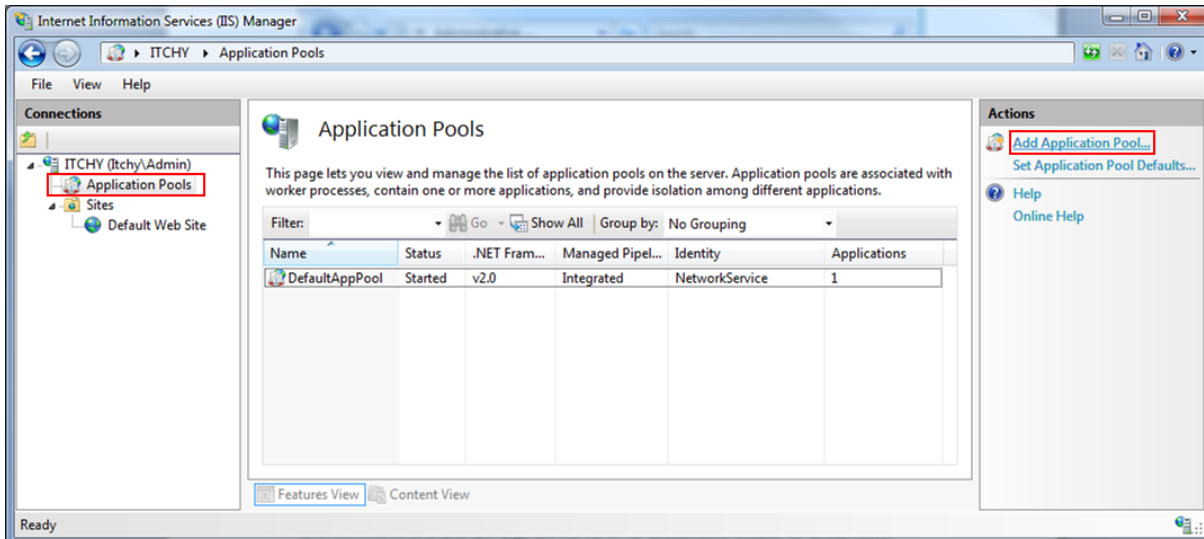
For Windows Authentication to work, the account set for the application pool identity (in the application pool’s advanced settings) must have access to the VETtrak SQL Server database.

## 1.3 Step 3 - Adding an Application Pool

Open the IIS Manager from **Control Panel --> Administrative Tools --> Internet Information Services (IIS) Manager**.

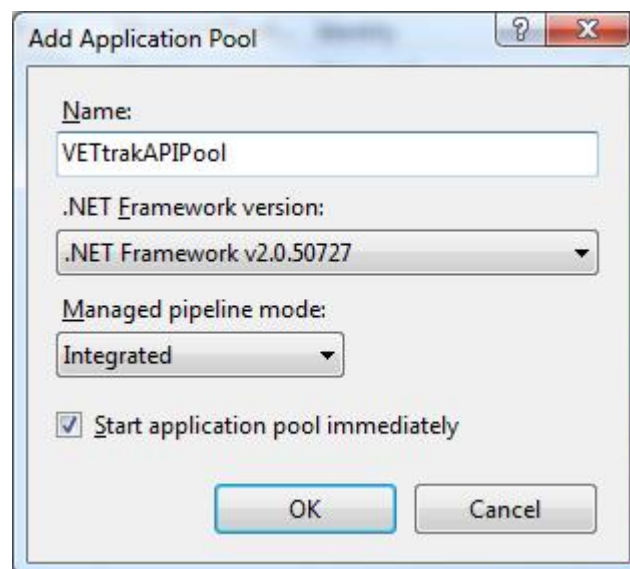
It is recommended that the APIs be set up as an Application running within its own Application Pool. This isolates the process and allows the APIs to be managed independently of other IIS applications running on the server.

To do this, in the IIS Manager, click on **Application Pools** in the tree in the left panel, and click **Add Application Pool** on the right.

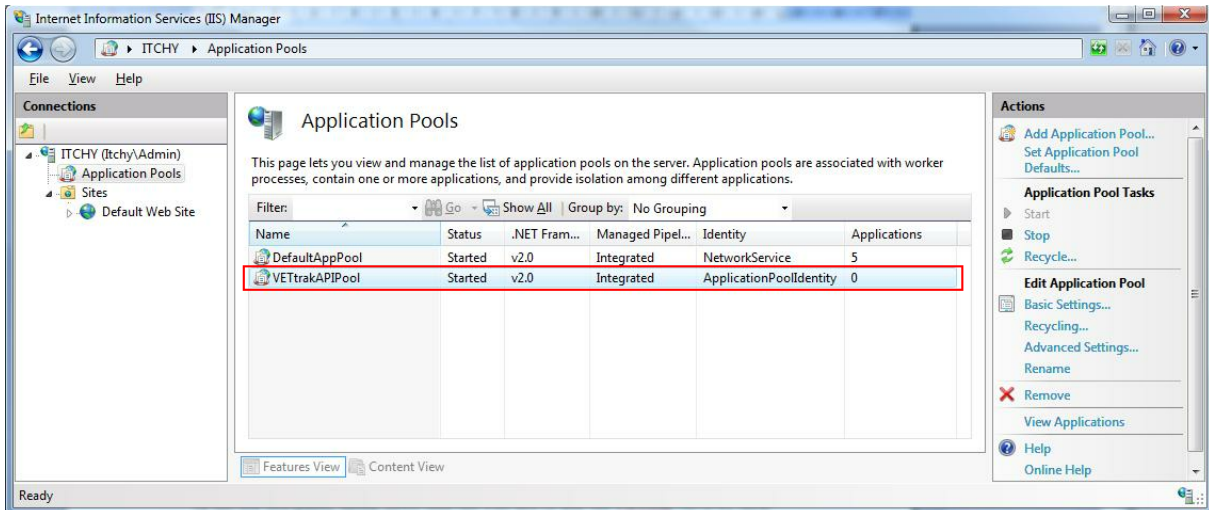


In the **Add Application Pool** window, enter the following, followed by **OK**:

- **Name:** VETtrakAPIPool
- **.NET Framework version:** A version starting with “.NET Framework v2”
- **Managed pipeline mode:** Integrated
- **Start application pool immediately:** Ticked

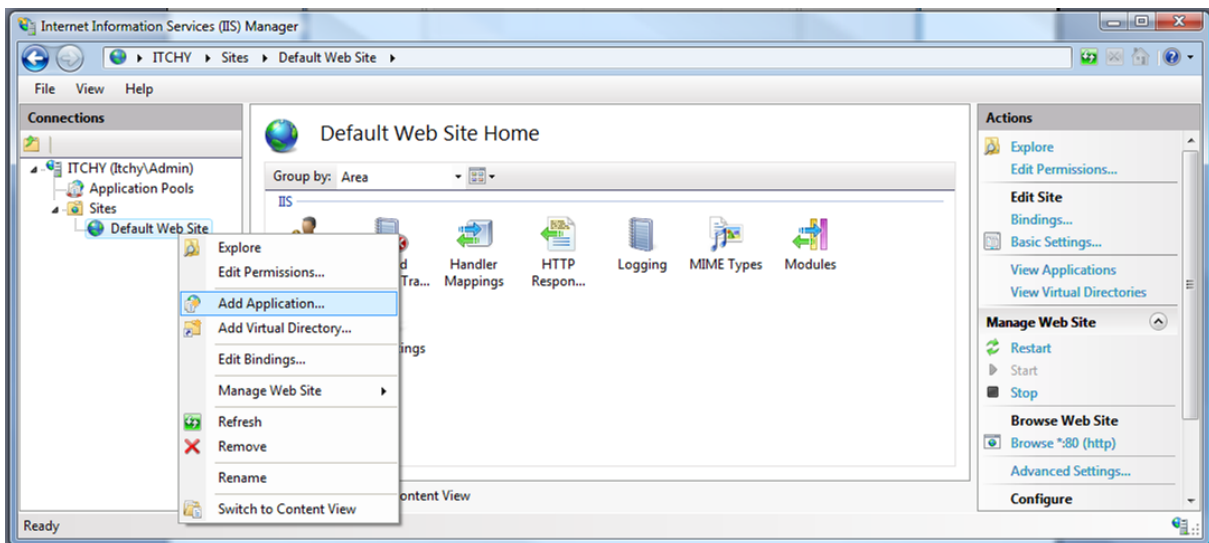


This will add the Application Pool to the IIS Manager:



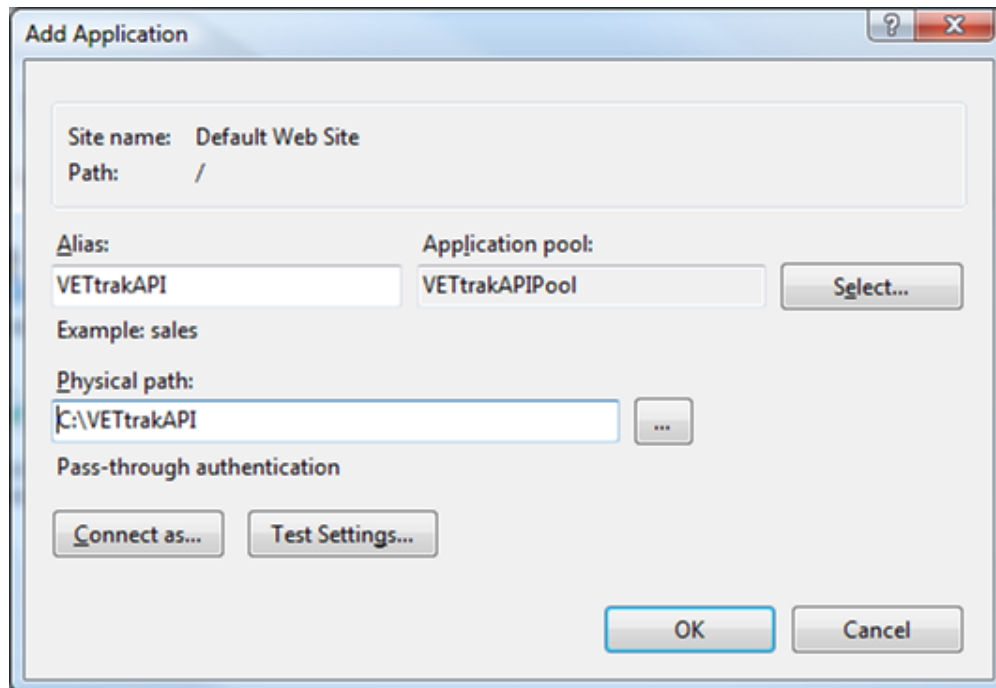
## 1.4 Step 4 - Add Application

In the left tree panel, under **Sites**, find the web site to add the VETtrak APIs to, such as the **Default Web Site** (a new web site may be created first if desired). Right-click on this web site and click **Add Application**. Note that if the VETtrak APIs were copied to a folder under the web root, it will already appear as a folder under the web site (you may need to refresh in order to see it) – in this case, instead of adding the application, right-click on the folder and select **Convert to Application**.

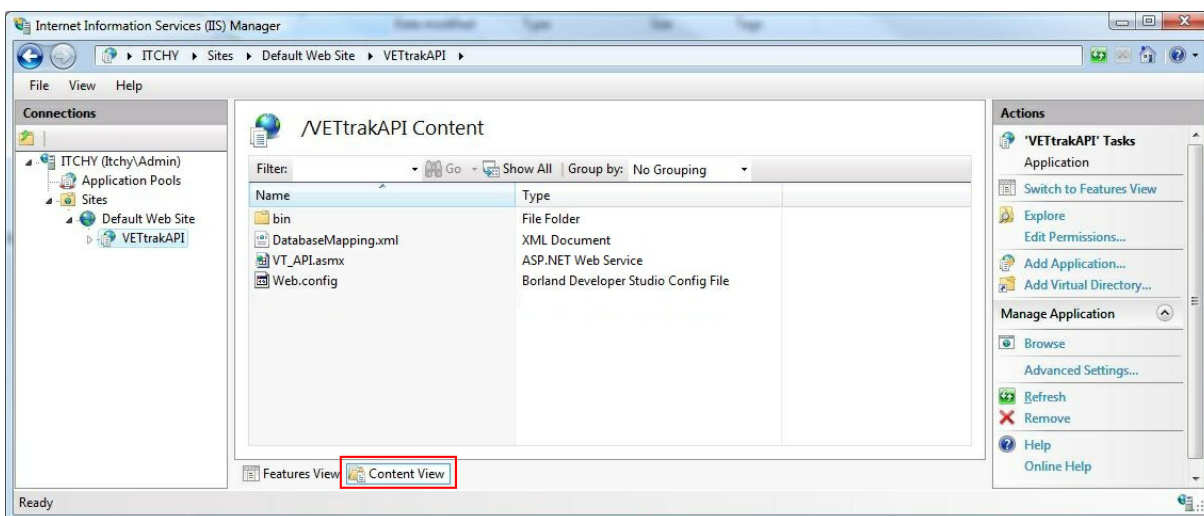


In the **Add Application** window, enter the following, and click **OK**:

- **Alias:** This is what the URL to the APIs will be. This can be whatever you want. For the purposes of this document, we will use "VETtrakAPI"
- **Application pool:** Click the **Select** button and choose the VETtrakAPIPool application pool from the list
- **Physical path:** Use the ... button to browse for and select the folder into which the VETtrak API ZIP was extracted. For the purposes of this document, a path of "C:\VETtrakAPI" is used



Within the new application, click the **Content View** button at the bottom to ensure you can see the asmx file(s) for the API(s) you have, the web.config file and the bin folder containing the DLLs.



## 1.5 Step 5 - Test that the VETtrak API is Accessible

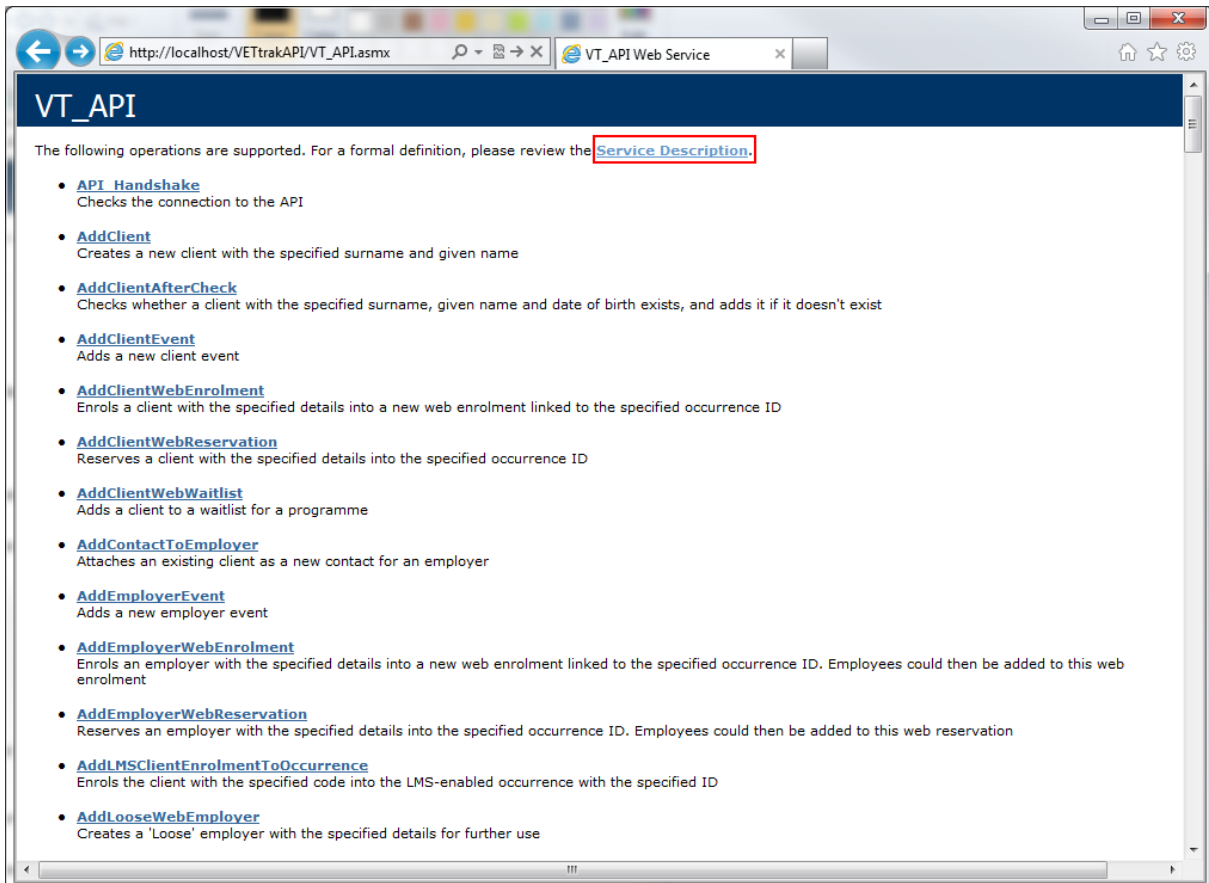
To test that the API is accessible, open an Internet Explorer browser, and enter the URL for the API:

[http://host\\_name/application\\_name/VT\\_API.aspx](http://host_name/application_name/VT_API.aspx)

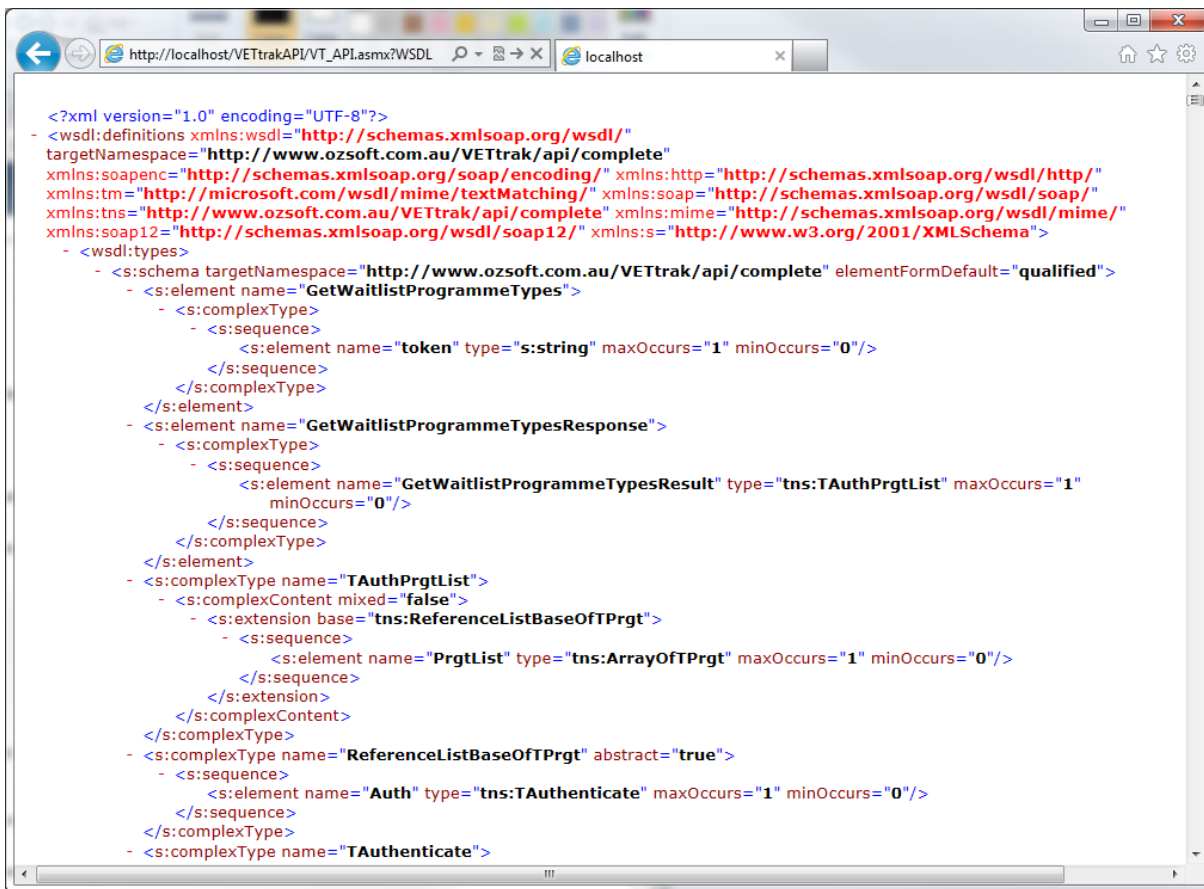
eg. For the example used in this document, the URL for the API will be:

[http://localhost/VETtrakAPI/VT\\_API.aspx](http://localhost/VETtrakAPI/VT_API.aspx)

If everything is set up correctly, a service information page should appear, like the following:

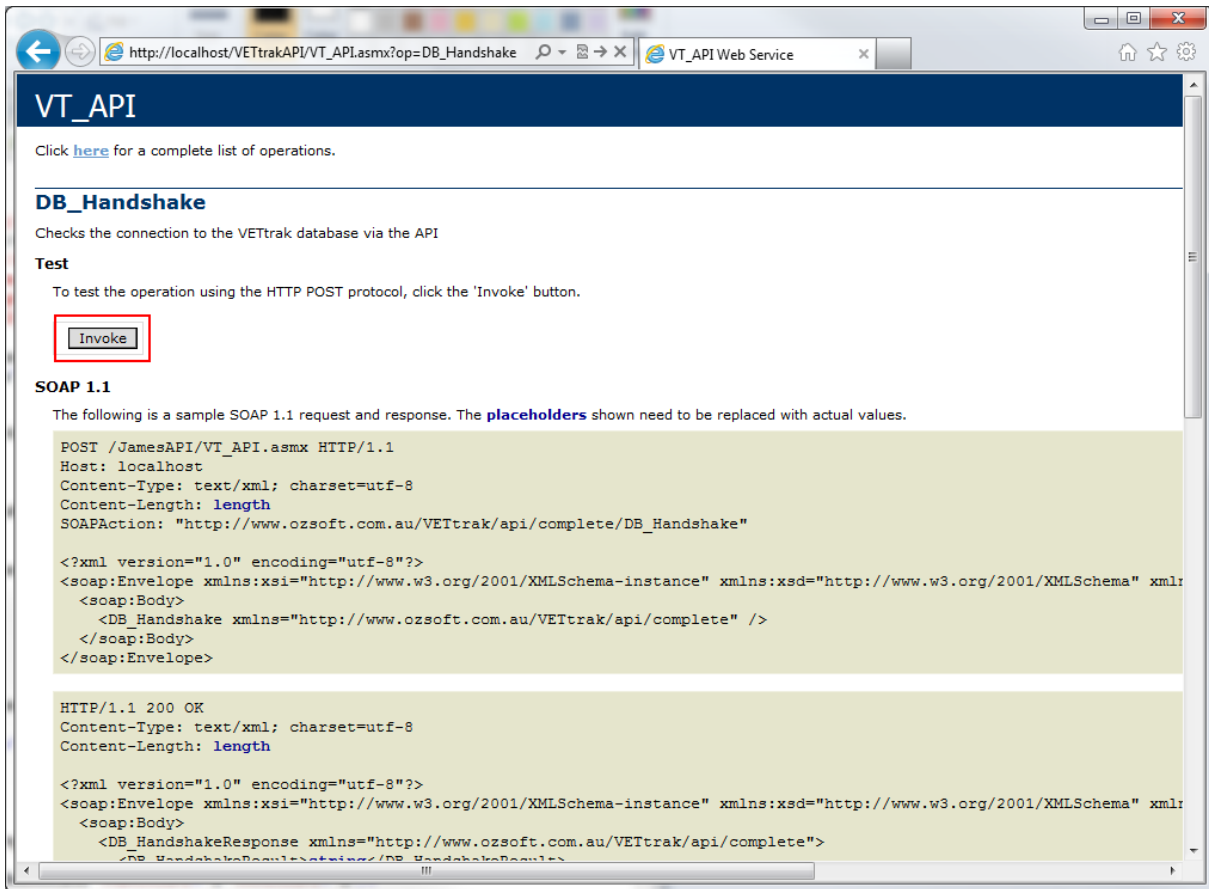


Click on the **Service Description** link to view the web service definition XML (WSDL):

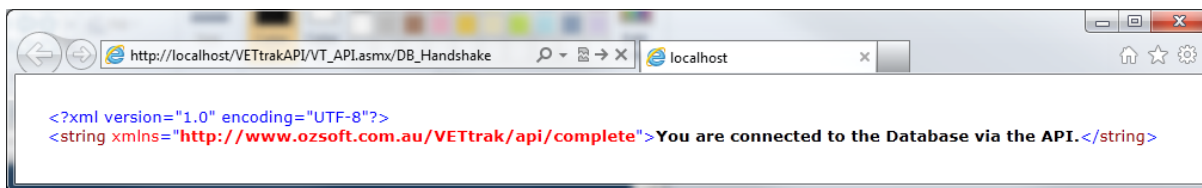


```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.ozsoft.com.au/VETtrak/api/complete"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.ozsoft.com.au/VETtrak/api/complete" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:s="http://www.w3.org/2001/XMLSchema">
  - <wsdl:types>
    - <s:schema targetNamespace="http://www.ozsoft.com.au/VETtrak/api/complete" elementFormDefault="qualified">
      - <s:element name="GetWaitlistProgrammeTypes">
        - <s:complexType>
          - <s:sequence>
            <s:element name="token" type="s:string" maxOccurs="1" minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      - <s:element name="GetWaitlistProgrammeTypesResponse">
        - <s:complexType>
          - <s:sequence>
            <s:element name="GetWaitlistProgrammeTypesResult" type="tns:TAuthPrgtList" maxOccurs="1"
minOccurs="0"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      - <s:complexType name="TAuthPrgtList">
        - <s:complexContent mixed="false">
          - <s:extension base="tns:ReferenceListBaseOfTPrgt">
            - <s:sequence>
              <s:element name="PrgtList" type="tns:ArrayOfTPrgt" maxOccurs="1" minOccurs="0"/>
            </s:sequence>
          </s:extension>
        </s:complexContent>
      </s:complexType>
      - <s:complexType name="ReferenceListBaseOfTPrgt" abstract="true">
        - <s:sequence>
          <s:element name="Auth" type="tns:TAuthenticate" maxOccurs="1" minOccurs="0"/>
        </s:sequence>
      </s:complexType>
      - <s:complexType name="TAuthenticate">
```

To test the API's connectivity to the database, back on the **Service Information** page, click the **DB\_Handshake** link in the list:



Then click the **Invoke** button (by default, this button will only appear when using a browser on the same machine the API is installed on, but this can be changed in the web.config file). The result should be an XML string with a success message:

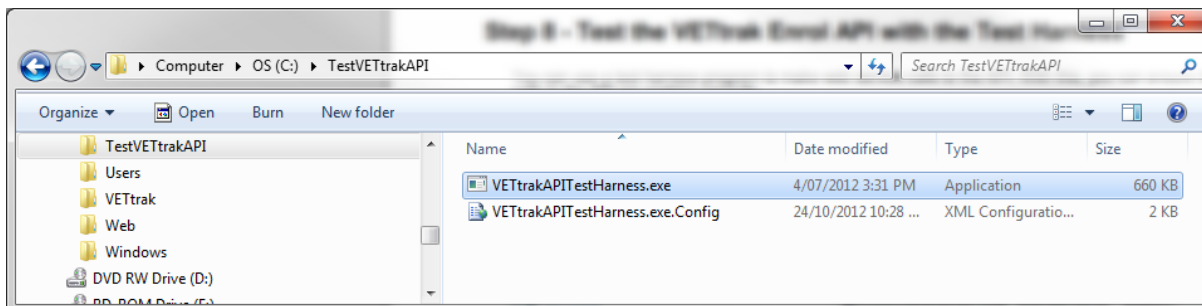


If the result is an error message, review the error and correct the connection string in the web.config file.

## 1.6 Step 6 - Test the VETtrak Enrol API with the Test Harness

You can use a test harness program to make web service calls to the API. With this, you can ensure that SOAP calls to the API are working correctly.

To use the VETtrak API test harness, copy the **VETtrakAPITestHarness.exe** and **VETtrakAPITestHarness.exe.config** files from the VETtrakAPITestHarness folder in the VETtrak ASP.NET API ZIP file to a new folder (preferably on a different machine, which must have .NET Framework 3.5 installed).



Double-click the VETtrakAPITestHarness.exe to run it.



**VETtrak API Test Harness**

**Connection**  
 Server address (eg. http://192.168.1.1/VETtrakAPI/VT\_API.asmx)  
 http://localhost/VETtrakAPI/VT\_API.asmx **Init OK**  
 Username Password Token  
 vettrak trakker **Authenticate** \$-7&)8.\"/>

**GetPaymentTypes: Retrieved 5 payment types**  
 10:58:08  
 GetLocations: Retrieved 21 locations  
 10:58:13  
 GetClientExtendedDetails: Retrieved details for client code 00525.  
 2 enrolments, 3 contracts, 1 awards, 2 classes, 45 units, 12 events.

Retrieved Data Start Date Retrieved Data End Date  
 1/01/1900 31/12/2100

**Web Enrol Client Staff Employer LMS Occurrence Qualification Reference Query Data SMS**

Client Code 00525 **Retrieve Client** Given Test Surname Person

**Find Create Details AVETMISS Enrolments Contracts Awards Classes/Attendance Units/Results Events Staff Invoices Tasks**

Title Given name Surname Other name Gender  
 Mr Test Person M

Date of Birth Email Emergency Contact Name  
 22/02/1981 here@there.com Mr Bloggs

Mobile phone Home phone Work phone Fax phone Emergency Phone Address Type  
 0400112233 0366554433 0311223344 0399887766 Normal

Residential address Postal address  
 Address Address  
 25 There St 31 Here St  
 City Postcode State City Postcode State  
 Rouse Hill 2155 NSW Rouse Hill 2155 NSW

Employer Identifier Position Commenced Last assessed Terminated  
 35 Operations Manager 24/10/2012 24/10/2012 24/10/2012

Visa Number Visa Type Visa Expiry Passport number Salary  
 24/10/2012 0

Update additional data

Username Password  
 ahmd1 3S3vxX **Update username/password**

**Update Build XML Update**

In the **Server address** field, enter the URL of the VT\_API.asmx. If you are running the test harness on the same machine the API is installed on, this will be the same URL that was used in step 5. Click the **Initialise** button. If this is successful, the button will change to **Init OK**.

Enter the Username and Password to access VETtrak (such as the default vettrak/trakker), and click the **Authenticate** button. If successful, a token appears.

The other tabs and buttons can then be used to test various functions in the API. This includes:

- The **Web Enrol** tab for online enrolments, including loading and viewing web programmes and occurrences, creating and updating clients and employer web enrolments, and recording payments
- The **Client**, **Staff** and **Employer** tabs for portals functionality, including viewing various details of clients, staff members and employers, and updating some details
- The **LMS Occurrence** tab for LMS integration functionality, such as viewing and updating details of LMS published occurrences and enrolments