

CCB NZ B2B Web Service Debugging

For debugging remote and local CCB NZ B2B web services.

Test Harness

The CCB NZ B2B Web Service can be debugged on a local system using the Windows Forms test harness application specifically designed and developed for it. The application file is xxx.exe and the title of the application is CCB Web Service Test Utility. It is configured with the xxx.xml configuration file. There may be other configuration files available that can be used for different environments, e.g.:

xxx.xml
xxx.xml
xxx.xml
xxx.xml

The above configuration files refer to test, production, etc. To use any of these configuration files, simply re-name the file to xxx.xml after backing up the original and re-start the test harness application.

Once the application is running, the values in the configuration file can be confirmed by using the menu items Setup > Web Service and Setup > Subscriber. The forms should present values from the xxx.xml file and any modifications made in the forms will be saved to the xxx.xml file.

The solution and source code for this application can be found at the following location:

/xxx/

However pre-compiled application, configuration and sample files can be obtained from a CCB developer or tester.

Remote Debugging Configuration

Remote debugging refers to using the test harness to debug remote web services that exist in test, UAT and other such environments. For this example, the test environment would be used, which is available in the following configuration file:

xxx.xml

So ensure that this file is used as the configuration for the test harness as described above.

When debugging the main things to check prior to starting are, the URL of the web service that would be used (v1 Schema, Web service URL) along with the product code (Request product code) to use for the debugging, by using the Setup > Web Service menu item. For New Zealand only the v1 Schema, Web service URL is used and for Australia both the v1 Schema, Web service URL and the v2 Schema, Web service URL are used.

Some example product codes include xxx for New Zealand and xxx for Australia. The Environment field in this form has two possible values xxx for testing and xxx for production, a check to ensure that the correct remote web service is being used.

The other values to check prior to debugging are the subscriber values which can be accessed from Setup > Subscriber.

Remote Debugging

The web service request data can be entered into the main forms from tabs such as Personal Details, Enquiry Details / Application Scorecard, etc. However it is easier to use XML request files that can be loaded from File > Load Request ..., e.g. xxx.xml for use with the New Zealand web services. Once one of these XML files is loaded, it will populate the fields in the main form. Such existing files can be used and new ones can be also be created by using the File > Save Request commands after the main form values are populated and modified.

The New Zealand specific configurations to check from the main form are the Country which should be NZ, the Consumer/Commercial code is usually defined by the XML request that is loaded and the Schema Version which should be V1. The Consumer/Commercial code is 1 for Consumer and 2 for Consumer followed by Commercial (both).

Once the test harness is configured and the main form is populated with values, the SUBMIT button can be pressed to send the request to the web service. After a short while, a Response Viewer window will show with tabs displaying the response XML, HTML and any errors and warnings. The XML View shows the XML response from the web service, the HTML View shows the HTML representation of the XML response and the errors and warnings shows any errors that occurred.

CCB B2B Server Architecture

The CCB NZ B2B web service is represented in the xxx diagram as xxx:

xxx

It is also available in the xxx page:

xxx

In this page it can be searched for by its name xxx. From the xxx diagram, the xxx represents the test harness application for testing purposes.

In a production environment the xxx would use the xxx as shown in the architecture diagram. xxx is an IIS web services application located in the xxx. This means that it is open to the public and any user can access it. The rest of the system is protected by firewalls.

This web service receives requests and then re-directs those requests to either the xxx or xxx internal service applications, that are protected by firewalls, depending on where the request should go. xxx is a modern, Windows Services and WCF based application whereas xxx is an older HTTP SOAP XML based web service.

The XML schemas used to communicate between the xxx and the xxx are the external schemas for requests and responses. The XML schemas used to communicate between the xxx and the xxx or xxx are the internal schemas for requests and responses.

For xxx there is one internal schema and two external schemas. The xxx does the mapping between external and internal schemas. The xxx internal service can only understand the internal schema. There could be multiple external schemas for xxx, currently there are V1 and V2, but they all map to a generic internal schema that the xxx internal service can understand.

Since the xxx internal service application is older, there is no translation between the external and internal schemas at the xxx. So whatever requests and response that occur between the xxx, xxx and the xxx internal service application are the same, i.e. there is no translation between schemas during communication.

Normally for local debugging of the xxx and xxx internal service applications, the debugging of the xxx is also required. However since there is no translation of requests and responses to and from the xxx and the xxx internal service application, only the xxx internal service application need to be debugged with the xxx, which is represented by the test harness.

Local Debugging Configuration

Local debugging refers to using the test harness to debug local web services, usually available in local development environments. The solution to load in Visual Studio is:

xxx.sln

In the web.config of the ccbb2b web project, the following value exists:

```
<add key="xxx" value="xxx.pfx" />
```

Ensure that the file xxx.pfx exists at this path or modify the above value to the path where it can be found on the local system. This is a certificate that protects communication between this application and the external Scoring Engine application. DISE stands for Dun & Bradstreet Integrated Scoring Engine.

There is another value:

```
<add key="xxx" value="xxx" />
```

This is for a Windows Registry value that has extra configurations. This value can be found in the Windows Registry path on the local system:

xxx

In this path there is the value ConfigPath which is a path to the local system. This path should contain the location of the xxx.config file. The NZ applications use older remoting features and this configuration file is for that purpose. Multiple applications can use this file.

In this path, there is also a Connections folder that has values related to database connections. The idea is to have a central location that multiple applications can refer to for database connection information. If these settings are not there, the

application could cause exceptions, even if the connections have been defined in the web.config.

There are two Windows Registry configuration files that can be run to configure the above values on a local system:

xxx.reg
xxx.reg

These files can be edited in any text editor such as Notepad. The first file is used by 64-bit applications and the second one is for 32-bit applications. For example, in a server environment, the applications would be 64-bit and would use the configuration information from the first registry file that are available in the Windows Registry path:

xxx

But any 32-bit applications, such as Visual Studio IIS Express, would use the configuration values from the second file found in the Windows Registry path:

xxx

The above files can be obtained from a CCB developer or tester.

The easiest way to do local debugging is by using IIS Express with Visual Studio and using IIS requires extra configuration.

In Visual Studio, set the xxx web project as the start-up project and xxx.aspx as the start-up page. In the xxx web project properties, under the Web section, Servers, IIS Express should be selected. The Project URL field should have a value similar to <http://localhost:xxxxx/> where the xxxxx represents the port number assigned on the local development environment. This value along with the web service page name need to be used to configure the test harness as shown earlier, i.e:

<http://localhost:xxxxx/xxx.aspx>

Open xxx.aspx.cs and set a break point within the method:

```
public xxx xxx(xxx)
```

Local Debugging

Start Visual Studio debugging. The browser should open and it will navigate to the web service URL above. It should show the following web service operation xxx.

Start the test harness application and go to the menu Setup > Web Service. The v1 Schema, Web service URL needs to be updated with the above value, after the xxxxx is replaced with the port number specific to the local development environment. Ensure that a valid product code is entered and accept changes. The Setup > Subscriber values should also be checked as previously mentioned.

Load a XML request file as shown earlier and then press the SUBMIT button. A short time afterwards the previously set break point should be hit in Visual Studio. Continue debugging and then the test harness should show the response XML, HTML and any errors and warnings as shown earlier.

Meeting Recording

There was a meeting held on the 03rd of June 2022 and this document is based on that meeting. The recording of this meeting is available at this network location:

xxx

It is in the folder named "2022-06-03 CCB NZ B2B Web Service Build, Deployment to Test and Debugging" at the above network location.