

Integrating with FastraxNetwork Web Services

Contents

Overview	1
Service URL and Schemas.....	1
Sample Application.....	1
More Information.....	5
Request Statuses.....	5
Reports.....	6

Overview

Background verification services are provided via our Fastrax applications: FastraxSelect, our web site (<https://www.fastraxselect.com>), and FastraxNetwork, our SOAP web services. This document describes integrating with the FastraxNetwork services, and is intended for a technical audience already familiar with consuming web services.

Service URL and Schemas

The URL of FastraxNetwork is:

<https://ws.fastraxonline.com/FastraxRequest/FastraxNetwork.asmx>

Navigating to that page displays the documentation for existing methods.

Xml schemas referenced by FastraxNetwork can be found at:

<https://www.fastraxonline.com/schemadocs/ReportSchemas/>

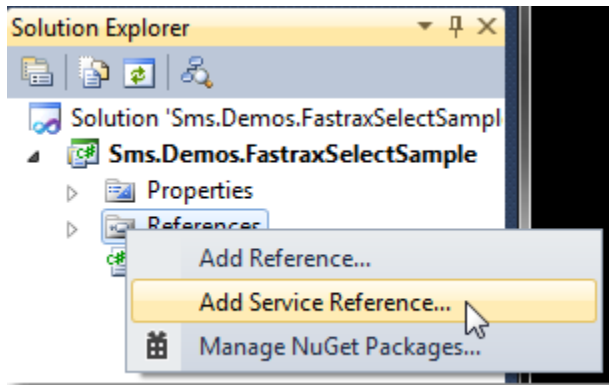
The schemas include:

Folder or File	Use
General Schemas	For searching requests
MainRequest.xsd	XML for any request. Contains elements for combined and individual reports.
MainResponse.xsd	The response to any MainRequest.
ReportSchemas	Schemas for report information used in MainRequests.
VendorSchemas	Used by researchers who integrate with Fastrax.

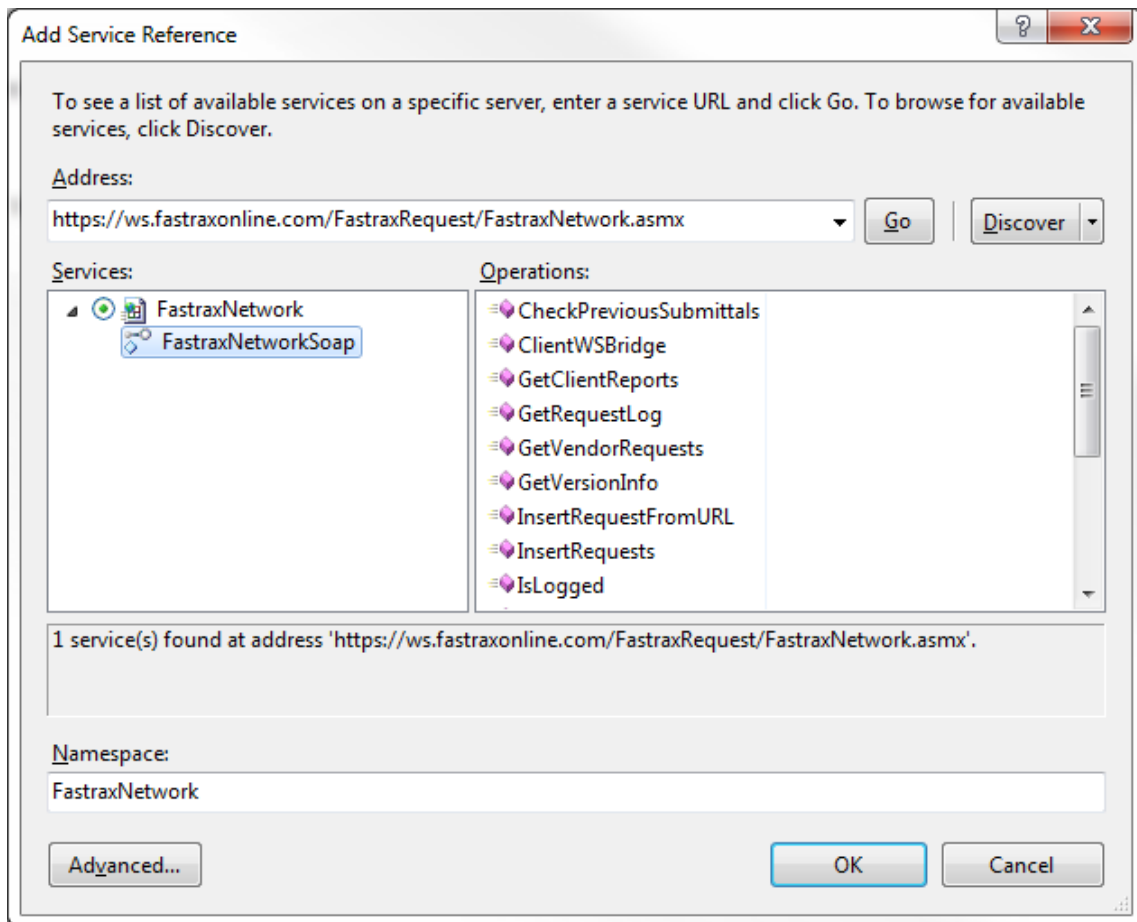
Sample Application

The following sample is written in Microsoft's C# language, using Visual Studio 2010. It demonstrates logging in, inserting a request for a Search America report, and retrieving the results for that request.

In Visual Studio, create a new Console project and open the Add Service Reference dialog. (Or, generate the classes using Microsoft's command line SvcUtil.exe tool.)



Enter the URL to the service, rename the Namespace, and click Go.



Use the following code in the Program.cs file.

```
using System;  
using System.Collections.Generic;  
using System.Linq;
```

```
using System.Text;
using System.Xml;
using Sms.Demos.FastraxSelectSample.FastraxNetwork;
namespace Sms.Demos.FastraxSelectSample
{
    class Program
    {
        static void Main(string[] args)
        {
            //Use your Customer Location and password for Credentials.
            string customerLocation = "YOURCUSTOMERLOCATION";
            string password = "YOURPASSWORD";

            //Create an instance of the web service.
            FastraxNetworkSoapClient fn = new FastraxNetworkSoapClient();

            //Get login instance.
            Console.WriteLine("Getting login information.");
            FastraxLogin login = fn.Login(customerLocation, password);

            //Create the request.
            string requestXml = @"<?xml version=""1.0"" encoding=""utf-8"">
                <FastraxRequests>
                    <request>
                        <subject-info>
                            <first-name>John</first-name>
                            <last-name>Doe</last-name>
                            <dob>1970-10-15</dob>
                            <ssn>999-99-9999</ssn>
                            <sex>M</sex>
                            <address-info>
                                <address>
                                    <street-number>1</street-number>
                                    <street-name>Main</street-name>
                                    <street-type>Str</street-type>
                                    <city>Cincinnati</city>
                                    <state>OH</state>
                                    <zip>45202</zip>
                                </address>
                                <phone-info>
                                    <phones>
                                        <phone-number>(513) 555-1234</phone-number>
                                        <phone-type>Cell</phone-type>
                                    </phones>
                                </phone-info>
                            </address-info>
                            <maiden-name>Deer</maiden-name>
                        </subject-info>
                        <report-info>
                            <individual-report ind-report-id=""24"">
                                <searchamerica />
                            </individual-report>
                        </report-info>
                        <client-field>
                            <id>Any info you want</id>
                            <our-trans>Echoed back to you in the report response</our-trans>
                        </client-field>
                    </request>
                </FastraxRequests>
            </?xml>
```

```
</FastraxRequests>";

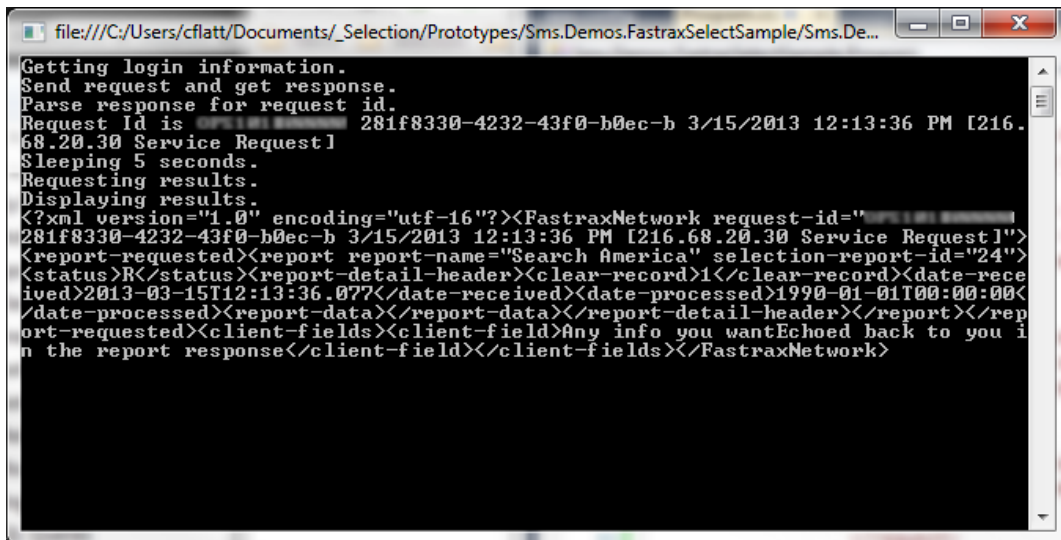
//New Request - look at the returned Xml doc to get information about the
//request, the key is the RequestID.
Console.WriteLine("Send request and get response.");
string responseXml = fn.InsertRequests(login, requestXml);

Console.WriteLine("Parse response for request id.");
XmlDocument xmlResponse = new XmlDocument();
xmlResponse.LoadXml(responseXml);
string requestId = xmlResponse.DocumentElement
    .SelectSingleNode("@//FastraxNetwork/request/@request-id").Value;
Console.WriteLine("Request Id is " + requestId);

//For this example, allow the real time report time to process.
Console.WriteLine("Sleeping 5 seconds.");
System.Threading.Thread.Sleep(5000);

//Format an Xml document, use a Request ID to search and retrieve results
//of the Request
string xmlString = @"<?xml version=""1.0"" encoding=""utf-8"" ?>
    <FastraxNetwork><search-criteria><request-id>"
        + requestId + "</request-id></search-criteria></FastraxNetwork>";
Console.WriteLine("Requesting results.");
string resultsXml = fn.RetrieveResults(login, xmlString);
Console.WriteLine("Displaying results.");
Console.WriteLine(resultsXml);
Console.ReadLine();
    }
}
}
```

Running the program with your credentials displays something like this:



```
file:///C:/Users/cflatt/Documents/_Selection/Prototypes/Sms.Demos.FastraxSelectSample/Sms.De...
Getting login information.
Send request and get response.
Parse response for request id.
Request Id is 281f8330-4232-43f0-b0ec-b 3/15/2013 12:13:36 PM [216.68.20.30 Service Request]
Sleeping 5 seconds.
Requesting results.
Displaying results.
<?xml version=""1.0"" encoding=""utf-16""><FastraxNetwork request-id=""281f8330-4232-43f0-b0ec-b 3/15/2013 12:13:36 PM [216.68.20.30 Service Request]"">
<report-requested><report report-name=""Search America"" selection-report-id=""24"">
<status>R</status><report-detail-header><clear-record>1</clear-record><date-received>2013-03-15T12:13:36.077</date-received><date-processed>1990-01-01T00:00:00</date-processed><report-data></report-data></report-detail-header></report></report-requested><client-fields><client-field>Any info you want Echoed back to you in the report response</client-field></client-fields></FastraxNetwork>
```

More Information

Once you have received your valid FastraxLogin object, you have a limited amount of time to make a transaction, after which your session will time out. You can get this time limit by looking at the EXPIRE member field of your FastraxLogin object. The measurement is in seconds.

The InsertRequests method sends report requests to FastraxNetwork for processing. The argument is an xml string that is formatted in accordance with the MainRequest.xsd schema. The return value is an XML string containing the results of your attempted transaction.

The SearchRequests method allows clients to search all their recent requests, passing in an XML string of search parameters defined in SearchRequestsByParam.xsd. An XML string is returned. Generally, an entire previous year's requests are available. Older requests can be retrieved by contacting customer service.

When submitting a request through the InsertRequests method, these values are used to specify which reports to include with the current request. Reports IDs are set as the ind-report-id attribute in the individual-report element.

Criminal County Report

```
<report-info>
  <individual-report ind-report-id="4">
    <criminalcounty>
      <location-info>
        <state>GA</state>
        <county>DeKalb</county>
      </location-info>
    </criminalcounty>
  </individual-report>
</report-info>
```

Search America Report

```
<report-info>
  <individual-report ind-report-id="24">
    <searchamerica></searchamerica>
  </individual-report>
</report-info>
```

Request Statuses

R = Received – request has been successfully submitted to our research network

P = Pending – we are currently researching this request

D = Done – all research has been done for this request but it has not yet been viewed by the client

C = Complete – the client has viewed the completed request at least once

Reports

Reports and their corresponding IDs are listed below. Your account may not be configured for all of these reports. If you would like to add additional reports, please contact your account manager.

Id	Name
12	Character Reference
8	Civil City
7	Civil County
6	Civil State
9	Credit
5	Criminal City
4	Criminal County
3	Criminal State
17	Drug Screen
10014	Drug Screen DOT
10013	Drug Screen DOT W/TS
10001	Drug Screen Hair 5
10345	Drug Screen Med A+MEP/FEN+COT
10003	Drug Screen Oral 6
10008	Drug Screen Urine 10
10002	Drug Screen Urine 10 + COTININEW/NIT
10012	Drug Screen Urine 10 + MDMA
10017	Drug Screen Urine 10 GC/MS
10009	Drug Screen Urine 10 W/OPI
10007	Drug Screen Urine 10/2000
10005	Drug Screen Urine 5
10015	Drug Screen Urine 5 GC/MS
10016	Drug Screen Urine 7 GC/MS
10010	Drug Screen Urine 7 W/NIT
10006	Drug Screen Urine 9
10241	Drug Screen Urine 9 + COTININEW/NIT
10004	Drug Screen Urine 9 + MDMA/6AM/A3
21	Education Verification
3474	Employment Eligibility Verification
14	Employment Verification
23	Federal Courts Report
4935	Fingerprint - FBI
4934	Fingerprint - OH
25	Insurance License Validation
11	Medical Sanctions
19	Motor Vehicle Report
4818	Name Trace Report
16	National Wants and Warrants
13	Professional License Verification
5000	Release Form
24	Search America
5001	Sex Offender Registry
20	SSN Trace Report
15	Workers' Compensation