RouteSavvy Webservice/API User Guide

Overview

The OnTerra RouteSavvy Optimization Service is a SOAP/XML web service that you can use with any application that supports SOAP/XML standards, including .NET, Silverlight, Java applications etc. It delivers sub-second optimization of up to a few hundred stops based on proximity, including different algorithms for (1) round trip, (2) one way with a defined end location and (3) one way with an undefined end location that will determine the optimal end location.

This service can be used in conjunction with routing services like Bing Maps to provide driving directions, route distance and route time. Visit http://msdn.microsoft.com/en-us/library/cc966826.aspx for more information on the Bing Maps routing service.

Sample Application

Following is a sample application (known as "RouteSavvy Online" – more at www.routesavvy.com) created using Silverlight and Bing Maps, showing before and after distance/time calculations. Please contact routesavvy@onterrasystems.com for more information. The application can be tried out at: http://online.routesavvy.com, after registering for a trial account at http://register.routesavvy.com.

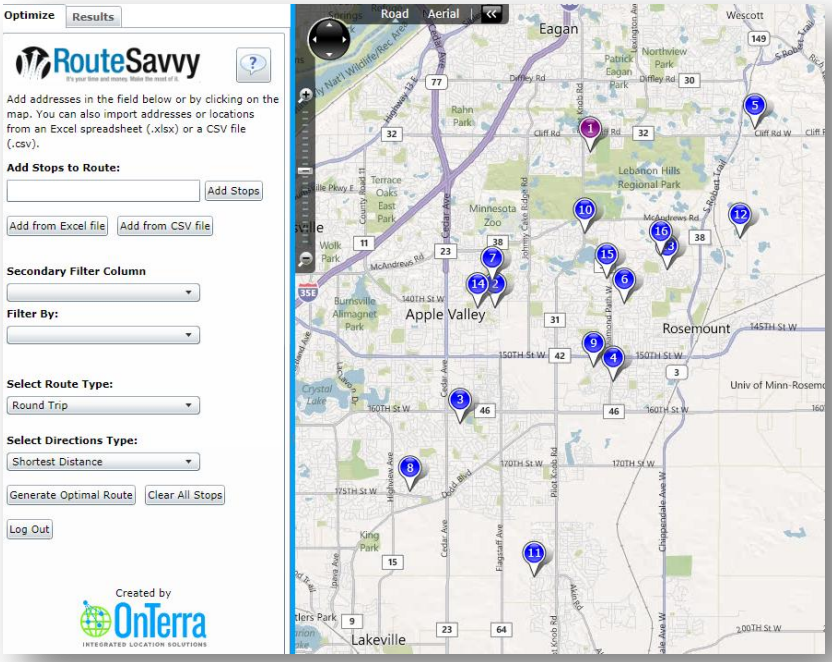


Figure 1: Unoptimized set of stops

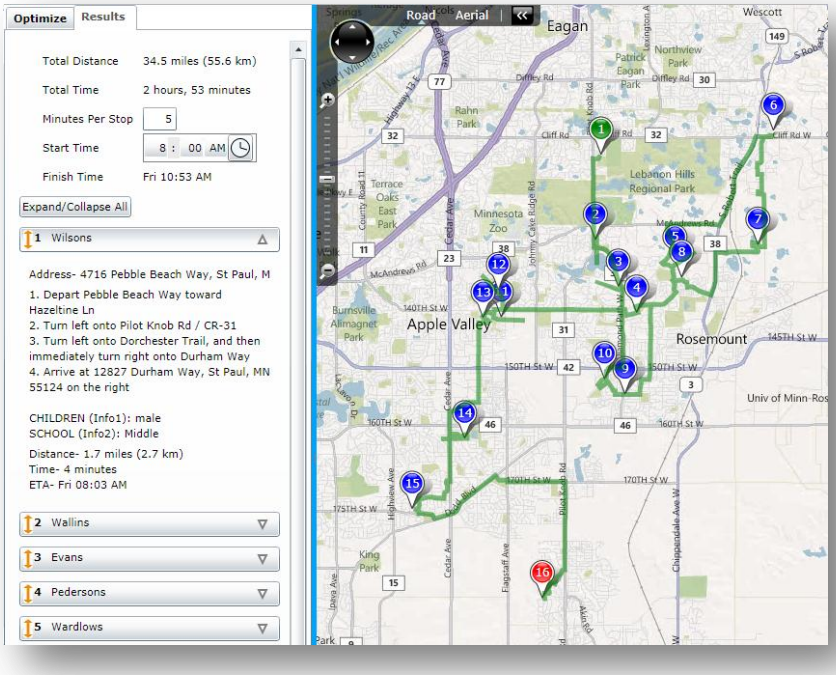
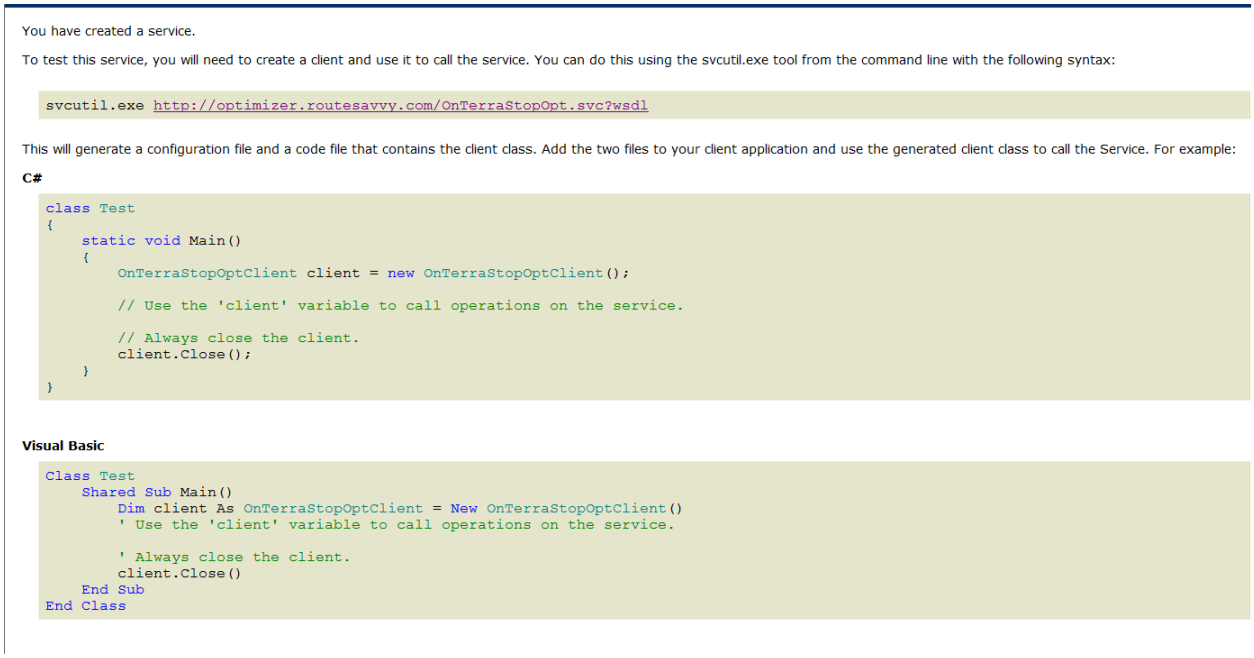


Figure 2: Optimized Route

Getting Started with .NET

Below are the steps to get started using the OnTerra RouteSavvy Optimization Service.

1) Add a reference to the production service:
   http://optimizer.routesavvy.com/OnTerraStopOpt.svc?wsdl



You have created a service.

To test this service, you will need to create a client and use it to call the service. You can do this using the svcutil.exe tool from the command line with the following syntax:

```
svcutil.exe http://optimizer.routesavvy.com/OnTerraStopOpt.svc?wsdl
```

This will generate a configuration file and a code file that contains the client class. Add the two files to your client application and use the generated client class to call the Service. For example:

C#

```
class Test
{
    static void Main()
    {
        OnTerraStopOptClient client = new OnTerraStopOptClient();

        // Use the 'client' variable to call operations on the service.

        // Always close the client.
        client.Close();
    }
}
```

Visual Basic

```
Class Test
    Shared Sub Main()
        Dim client As OnTerraStopOptClient = New OnTerraStopOptClient()
        ' Use the 'client' variable to call operations on the service.

        ' Always close the client.
        client.Close()
    End Sub
End Class
```

Figure 3: Production Web Service Page

1. Create Parameter Object and set value
   - Set the AppID
     Parameters.AppID = "<Your Token>" (Replace <Your Token> with the actual Token that you have received after signed-up)

   - Set the RouteType
     Parameters.RouteType.RoundTrip
     Parameters.RouteType.FixedStartFixedEnd
     Parameters.RouteType.FixedStartOpenEnd
     Parameters.RouteType.BestStartBestEnd

   - Set the StartLocation and EndLocation parameters.
     Parameters.StartLocation (Required for RoundTrip, FixedStartFixedEnd and FixedStartOpenEnd)
     Parameters.EndLocation (Required for FixedStartFixedEnd)

2. Create Location object and Add locations

Location[i].LocationName
Location[i].Latitude
Location[i].Longitude

3. Call the Stop Optimizer
   GetStopOpt(Location, Parameters)

Sample solution is available upon request. Please send request to routesavvy@onterrasystems.com.

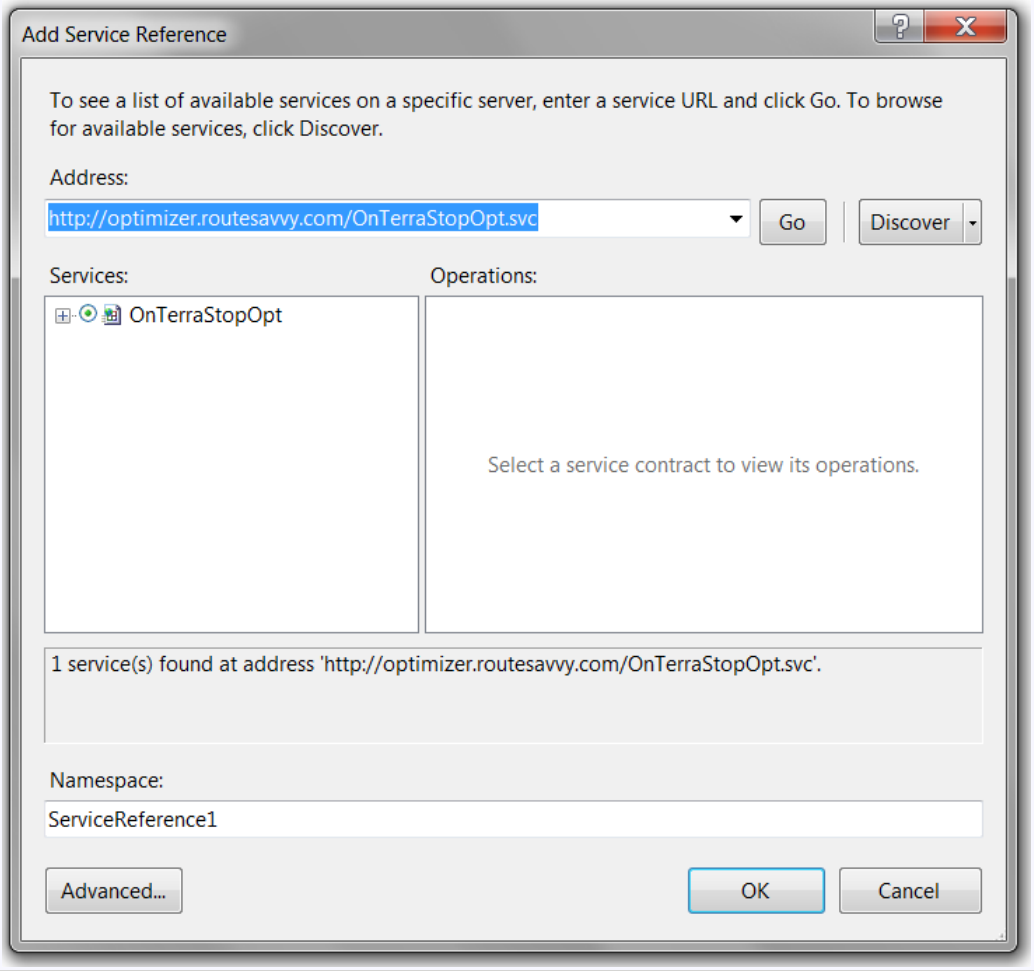
Code Walk-through

In this example, a DataTable with 3 columns (LocationReference, Latitude, Longitude) is created. Location information is read from the DataTable and the route is optimized.

Step 1: Create a simple windows forms application. Add a Button

```
<no code here>
```

Step 2: Add a service reference to OnTerra RouteSavvy service.



Step 3: Add the following references to the solution

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.ServiceModel;
```

Step 4: Create a method CreateDataTable(), and add some locations to the DataTable

```
DataTable table;
private void CreateDataTable()
        {
            table = new DataTable();

            table.Columns.Add("LocationReference", typeof(string));
            table.Columns.Add("Latitude", typeof(double));
            table.Columns.Add("Longitude", typeof(double));

            table.Rows.Add("7937 19TH AVE", 38.534297, -121.410634);
            table.Rows.Add("7921 AMADOR AVE", 38.537595, -121.41123);
            table.Rows.Add("8004 AMADOR AVE", 38.537175, -121.410121);
            table.Rows.Add("8008 BUTTE AVE", 38.53644, -121.409961);
            table.Rows.Add("7917 CARLTON RD", 38.538314, -121.411546);
            table.Rows.Add("8034 CLIFTON RD", 38.538602, -121.409643);
            table.Rows.Add("7901 MERCED AVE", 38.535414, -121.41101);

        }
```

Step 5: Call the CreateDataTable() method on Form Load Event

```
private void Form1_Load(object sender, EventArgs e)
{
            CreateDataTable();
}
```

Step 6: Include the following code in the button click event

```
private void button1_Click(object sender, EventArgs e)
        {
            OnTerraService.Location[] _Location = new
OnTerraService.Location[table.Rows.Count];
            for (int i = 0; i <= table.Rows.Count - 1; i++)
            {
                _Location[i].LocationName = table.Rows[i].ItemArray[0].ToString();
                _Location[i].Latitude = table.Rows[i].ItemArray[1].ToString();
                _Location[i].Longitude = table.Rows[i].ItemArray[2].ToString();
            }
```

```
            OnTerraService.RouteParameter _Parameters = new
OnTerraService.RouteParameter();
            _Parameters.AppId = "<Token>"; //Replace <Token> with the Token you
receive via email when you signed up
            _Parameters.RouteType = OnTerraService.RouteType.FixedStartOpenEnd;
            //Possible Route Types are:
            //OnTerraService.RouteType.RoundTrip
            //OnTerraService.RouteType.FixedStartFixedEnd
            //OnTerraService.RouteType.FixedStartOpenEnd

            _Parameters.StartLocation = "8004 AMADOR AVE";
            // This should be one of the LocationReference from the DataTable

            _Parameters.EndLocation = "";
            //EndLocation is needed only when the RouteType =
OnTerraService.RouteType.FixedStartFixedEnd;

            BasicHttpBinding binding = new BasicHttpBinding();
            UriBuilder serviceUri = new
UriBuilder("http://optimizer.routesavvy.com/onterrastopopt.svc");


            OnTerraService.OnTerraStopOptClient _OT = new
OnTerraService.OnTerraStopOptClient(binding, new EndpointAddress(serviceUri.Uri));
            OnTerraService.RouteOutput _Output = _OT.GetStopOpt(_Location,
_Parameters);
            if (_Output.ErrorCode == 0)
            {
                Console.WriteLine("The Optimized stop order is: ");
                for (int i = 0; i <= _Output.OutputLocations.Count() - 1; i++)
                {
                    Console.WriteLine(_Output.OutputLocations[i].LocationName + ", "
+ _Output.OutputLocations[i].Latitude + ", " + _Output.OutputLocations[i].Longitude);
                }
            }
            else
            {
                MessageBox.Show("Error " + _Output.ErrorCode + ": " +
_Output.OutputMessage);
            }
        }
```

Step 7: Run the application. Click the button and examine the output.

```
<no code>
```

Support

Please contact routesavvy@onterrasystems.com with details on your issue and we will respond as quickly as possible constrained only by the laws of physics and biology.

