Avaya Open Interfaces

CCMM Agent Web Services

AVAYA AURA CONTACT CENTER – CCMM AGENT WEB SERVICES

Introduction

CCMM Agent Web Services is a feature accompanying Avaya Aura Contact Centre. It is designed to support basic Email and functions on third-party clients (web or otherwise).

Supported configurations require that each agent be assigned to an email skillset and that each agent consume a multimedia license (as per the standard CC licensing model).

Supported functionality includes:

- Agent login
- Agent logout
- Read and Close contacts (based on IDs, available via CCT)
- Reply To / Reply All / Forward an open Contact
- Originate a new Email Contact
- Get a list of Closed Reason Codes
- Get a list of Skillsets
- Get All Agents.
- Get All Logged In Agents
- Transfer contacts to Agents/Skillsets

Getting Started

What is a Web Service Client?

A Web Service Client is any component or application that communicates with a web service using SOAP messages, or a comparable messaging protocol.

A Web Service Client can be a traditional client application. A client can also be another Web application. (In the latter scenario, the Web application would consume the XML within the SOAP message, format it, and send the result back to an ultimate client — perhaps a Web browser.)

We need these following five basic steps to create a Web service:

1. Create a proxy class for the Web service.
2. Reference the proxy class in the client code.
3. Create an instance of the proxy class in the client code.
4. Call the method on the proxy class that corresponds to the Web service method with which you want to communicate.

For most clients, these steps differ only in how the proxy class is referenced and how the Web service client is deployed. The tutorial below shows how to develop a simple Windows Forms application which logs into CCMM using CCMM Agent Webservices.

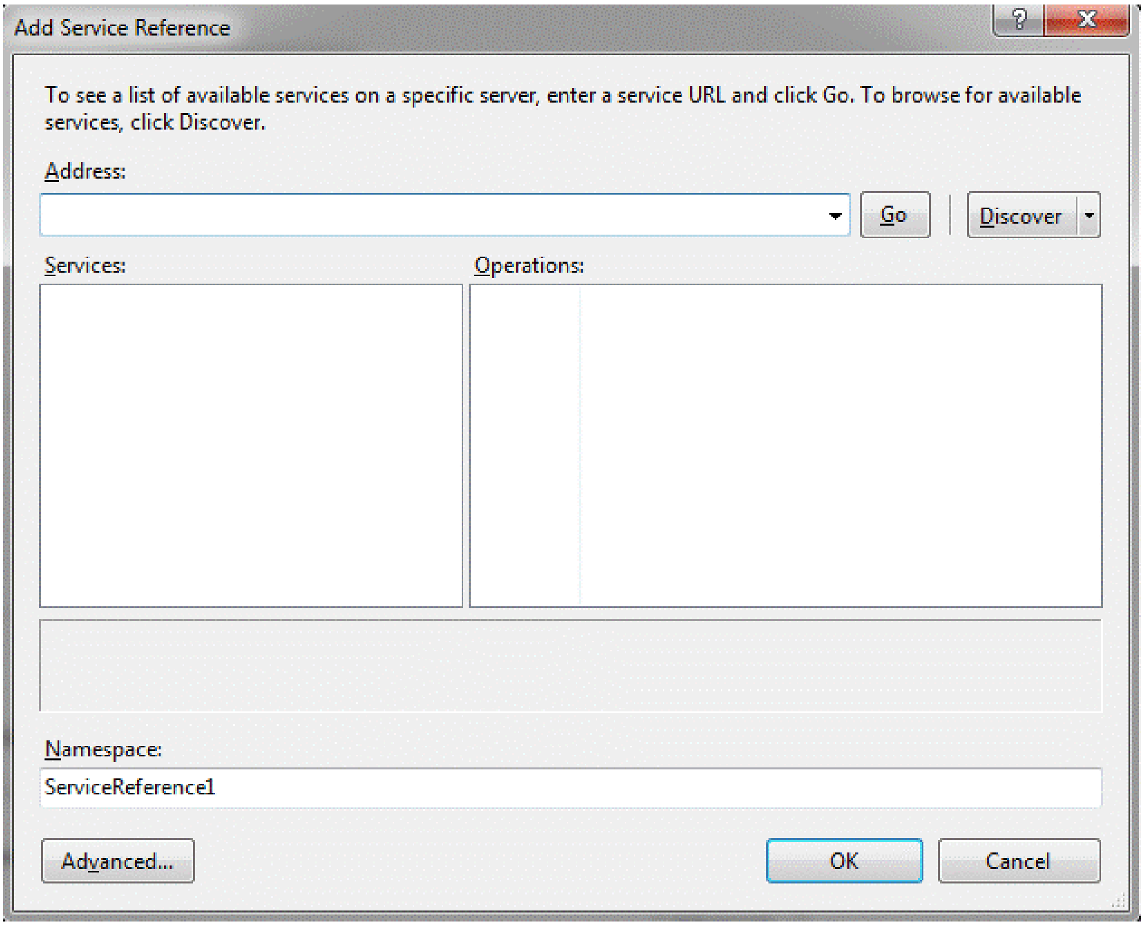
Using Visual Studio

Create a Windows-based application

1. Start Visual Studio 2008 or Visual Studio .NET.
   Note: Installing a Development Environment on a CCMM Contact Center Multimedia Server is not supported - please use a dedicated non-production machine.
2. On the **File** menu, point to **New**, and then click **Project**.
3. Under Project Type, click to select Visual C# Project.
4. Under Template, click to select Windows Form Application.
5. In the **Name** box, type **CCMMAgentWebservicesClient**, and then click **OK**.
   By default, a form named **Form1** is created.
6. Add a **Button** control and two **TextBox** controls to Form1.

Add a Reference to the Project

1. In Solution Explorer, right-click **CCMMAgentWebservicesClient**, and then click **Add Service Reference**.
   The **Add Service Reference** dialog box appears.

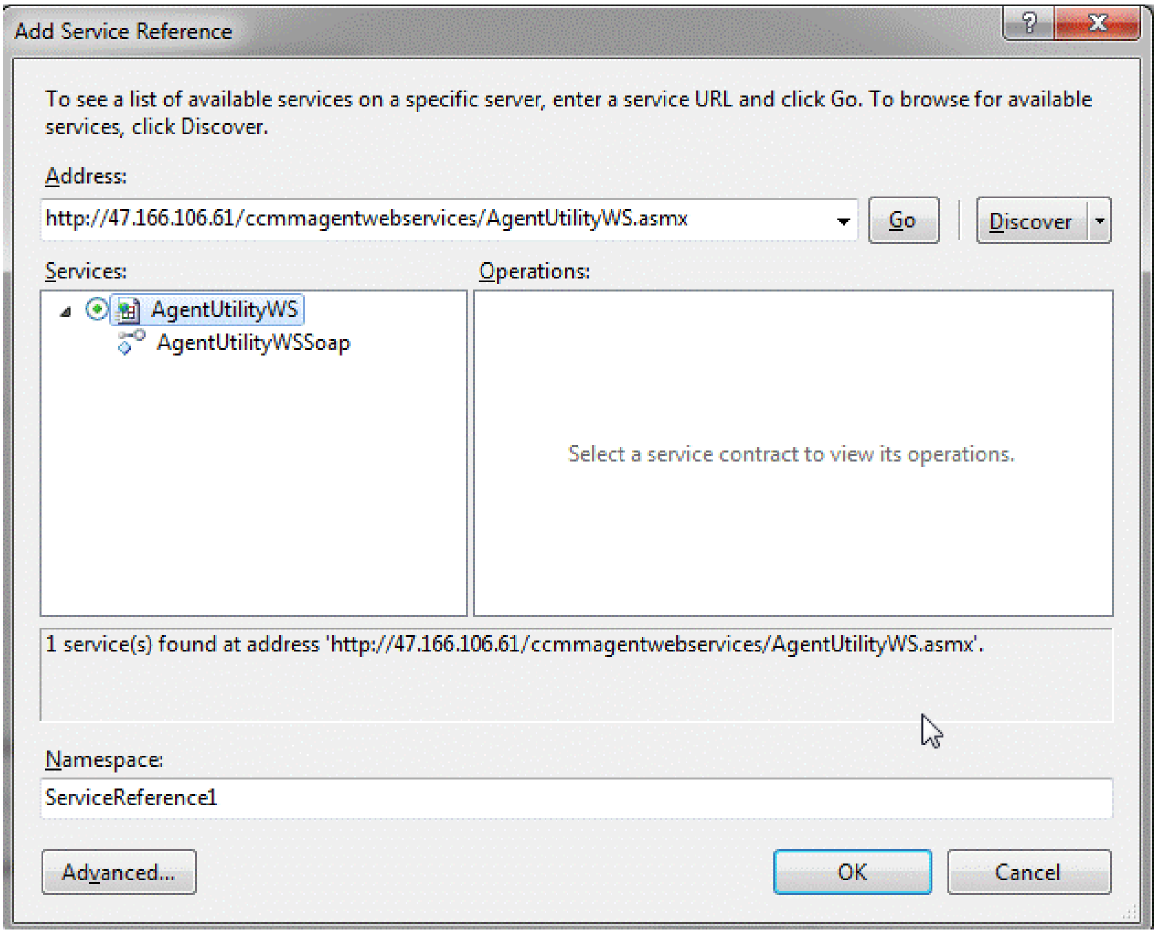


Add Service Reference
To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.
Address:
Go
Discover
Services:
Operations:
Namespace:
Advanced...
OK
Cancel

ServiceReference1

2. In the **Address** bar, type the URL of the Webservice ***http(s)://CCMMSERVERNAME/ccmmagentwebservices/AgentUtilityWS.asmx***.

   Use the correct protocol (HTTP or HTTPS) depending on the current Contact Center Web Services Security Level (On or Off).



To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

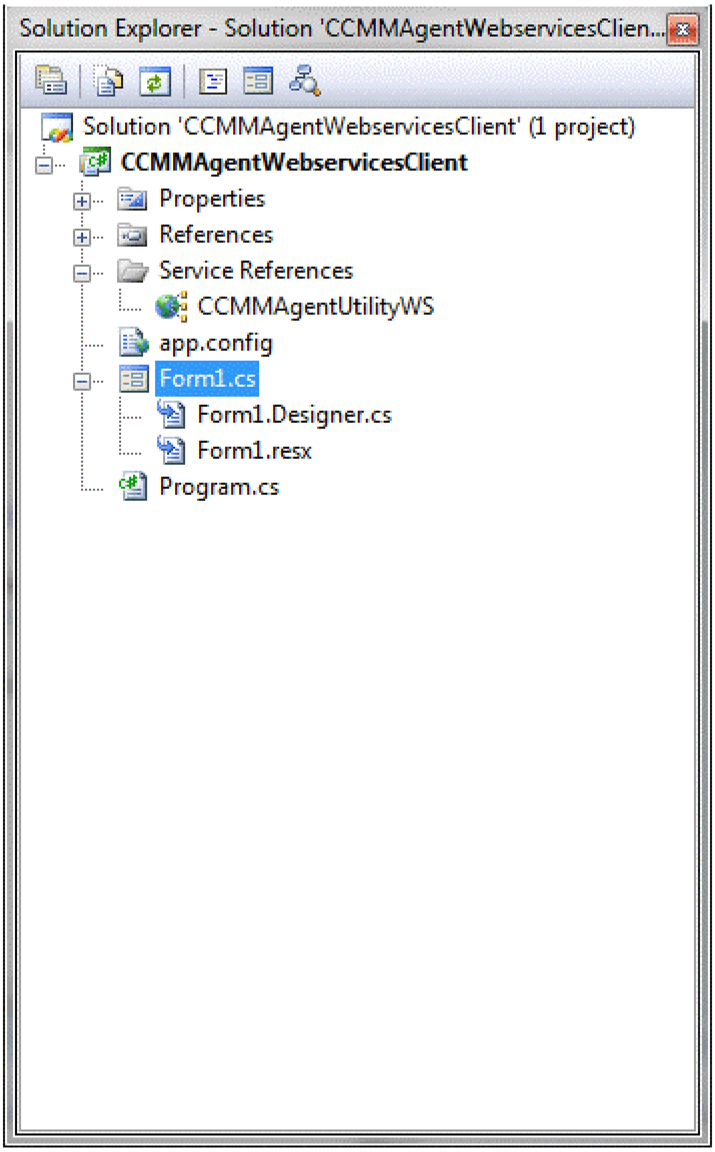
Address: http://47.166.106.61/ccmmagentwebservices/AgentUtilityWS.asmx

1 service(s) found at address 'http://47.166.106.61/ccmmagentwebservices/AgentUtilityWS.asmx'.

Namespace: ServiceReference1

Now click the Go Button to access the Web Service. In the Web service reference name space change the default name to **CCMMAgentUtilityWS**.

3. Press **Ok**.
   After the reference is added you will notice some new references in your Project.



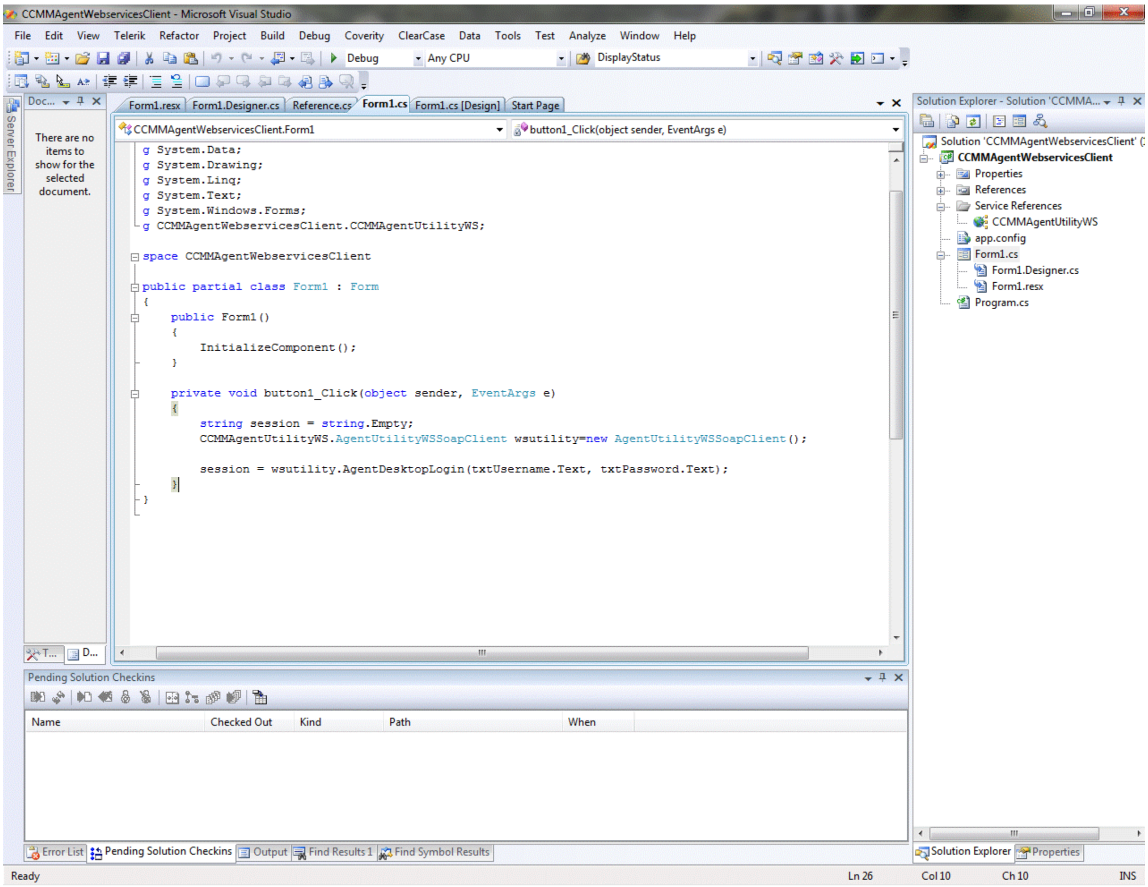
Solution Explorer - Solution 'CCMMAgentWebservicesClien...

Solution 'CCMMAgentWebservicesClient' (1 project)
- CCMMAgentWebservicesClient
  - Properties
  - References
  - Service References
    - CCMMAgentUtilityWS
  - app.config
  - Form1.cs
    - Form1.Designer.cs
    - Form1.resx
  - Program.cs

4. In Solution Explorer, right-click **Form1**, and then click **View Code**.
5. Add the following namespace at the top of the Form1.cs file.

   Using CCMMAgentWebservicesClient.CCMMAgentUtilityWS;

   Note: `CCMMAgentUtilityWS` is the name of the service reference as it appears under **Service References** in Solution Explorer. Double clicking on the CCMMAgentWebservicesClient under Service References will display the **Object Browser**. The name space here should correspond to the name space above (Using CCMMAgentWebservicesClient.CCMMAgentUtilityWS), if the two name spaces do not correspond change to the name space in code to the one under the object browser.


CCMMAgentWebservicesClient - Microsoft Visual Studio
File Edit View Telerik Refactor Project Build Debug Coverity ClearCase Data Tools Test Analyze Window Help
Debug
Any CPU
DisplayStatus
Form1.resx
Form1.Designer.cs
Reference.cs
Form1.cs
Form1.cs [Design]
Start Page
CCMMAgentWebservicesClient.Form1
button1_Click(object sender, EventArgs e)
There are no items to show for the selected document.
Solution Explorer - Solution 'CCMMA...
Solution 'CCMMAgentWebservicesClient'
CCMMAgentWebservicesClient
Properties
References
Service References
CCMMAgentUtilityWS
app.config
Form1.cs
Form1.Designer.cs
Form1.resx
Program.cs
Pending Solution Checkins
Name
Checked Out
Kind
Path
When
Error List
Pending Solution Checkins
Output
Find Results 1
Find Symbol Results
Solution Explorer
Properties
Ready
Ln 26
Col 10
Ch 10
INS

```
g System.Data;
g System.Drawing;
g System.Linq;
g System.Text;
g System.Windows.Forms;
g CCMMAgentWebservicesClient.CCMMAgentUtilityWS;

space CCMMAgentWebservicesClient

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string session = string.Empty;
        CCMMAgentUtilityWS.AgentUtilityWSSoapClient wsutility=new AgentUtilityWSSoapClient();

        session = wsutility.AgentDesktopLogin(txtUsername.Text, txtPassword.Text);
    }
}
```

6. Rename one of the TextBoxes we added earlier to ***txtUsername*** and the other to ***txtPassword*** - these will be used to send the username and password text to the AgentDesktopLogin Webservice.

   In the Form designer, double-click on the button we added earlier, and a click event will be created and attached to that button.

   Copy the code below under the click method - it will send the username and password supplied in the textboxes to the AgentDesktopLogin Webservice, and receive a valid SessionKey if login was successful. This SessionKey, a string, is needed for all subsequent calls to other CCMM Agent Webservices, as described in the API reference.

```
private void btnLogin_Click(object sender, System.EventArgs e)
{
string session = string.Empty;
CCMMAgentUtilityWS.AgentUtilityWSSoapClient wsutility=new
AgentUtilityWSSoapClient();
session = wsutility.AgentDesktopLogin(txtUsername.Text,
txtPassword.Text);
}
```

Further steps

Depending on the functionality you want to achieve, you can now invoke any of the Web Services in the CCMM Agent suite from your client application, formatting and displaying the results as per your requirements/design.

Please refer to the following elements within this SDK for further information:

- AACC 7.0 CCMM Agent Web Services README.txt.
- Sample client source code.
- AACC 7.0 CCMM Agent Webservices Help.chm – API documentation.