Metreos Communications Environment

Amazon Web Service Reference Application

8/16/2004
Version 1.0

OVERVIEW

The Amazon Web Service application bundled with the Metreos Communications Environment (MCE) is first and foremost intended for reference and educational purposes only. Additional features and error handling must be implemented before this application is suitable for use in a production environment.

This application demonstrates the considerations, flow, and behavior necessary to create a Cisco IP phone-based service. This sample will make use of HTTP provider-managed sessions, to allow an approach which should resonate with a web developer when using sessions found in ASP.NET or PHP, for example. A more thorough explanation of HTTP sessions can be found in the WebDialer documentation.

Along with sessions and the use of Cisco IP phone XML objects, this sample will make use of the rapid development and power behind web services, and show an example of incorporating web services into a native action.

The aim of the Amazon Web Service application is to allow a user with a Cisco IP phone to specify a search for Amazon products, and to then allow the user to see a more full description of the product if they so chose.

INSTALLATION

There are several components which must be configured in order for this application to function properly.

Media Server

The Media Server is not needed for this application.

Application Server

This guide assumes that the Application Server has been setup and configured according to the procedure outlined in the User's Guide. In particular, this application requires that the HTTP protocol provider be installed.

Start the Application Server.

The application must now be installed. This is done by clicking on the *Applications* link on the MCE Admin main page. Clicking *Add Application* will bring up a file chooser dialog for files with a *.mca* extension. Locate the bin/mca directory and double-click on *AmazonWebService.mca*. This will initiate the application installation procedure. The Application server will log a series of Info-level messages ending with an "installation successful" message.

Cisco CallManager

In CallManager 3.x or 4.x, a Cisco IP phone service must be set up to the URL which initiates an instance of this application, i.e., http://<ApplicationServerIP>:8000/AmazonWebService/MainMenu.

Once this is successfully completed, register a Cisco IP phone with this service for testing.

HOW IT WORKS

Overview

A Cisco IP phone sends HTTP requests based on user action. Given that we have constructed our application correctly, if the user presses a SoftKey, an HTTP request is sent to the script instance governing the current session. Multiple instances are created as multiple users concurrently use the Amazon Web Service application, so a requirement for correct behavior of this application is that the phone, when making requests, will signify that it is making a session-bound request. This behavior is made possible by instructing the phone to include a 'metreosSessionId' query parameter in every request to the Application Server. The Application Server will test for the presence for this query parameter, and route it to the script instance which matches its value.

The Cisco IP Phone has a number of predefined modes which this application makes use of. These modes are triggered by sending the phone certain XML-formatted data containing the appropriate information to active the desired mode.

The modes which the application uses are:

- CiscoIPPhoneMenu – lists products returned from search.
- CiscoIPPhoneText – shows a full description of a single item and displays error messages.

The Cisco IP phone is expecting properly formatted XML for each of these modes; here we make use of native actions found in the MCE framework which make creating these XML objects a simple matter for the developer, in that understanding the XML schema or even XML itself, is not necessary.

The flow of this application is standard for Cisco IP Phone service applications: the application is registered to handle various incoming HTTP requests, based on the action taken by the user.

In interacting with the web service provided by Amazon, we used the web services tools built into .NET and utilized by Microsoft Visual Studio.

Native Actions

The Weather Application uses 2 custom actions:

- SearchByKeyword – a generic keyword request is made based on the product line chosen and the search phrase entered.
- GetAllStations – a single item request is made based on the ASIN, which is the 'Amazon Standard Item Number'. Every item that Amazon sells has this number,

which we use to get the information for this item, and then format the information into a text string.

Native Types

No native types for this application

CLOSING REMARKS

Two additions to the Amazon Web Service application stand out as needed.

One would be to create paging for search results. Currently, only the first 10 results are shown.