

Test Hotlinks with a Web Browser

Document ID: 71485

Introduction

Prerequisites

Components Used

Conventions

Common Questions

How can I test Hotlinks with a web browser?

NetPro Discussion Forums – Featured Conversations

Related Information

Introduction

This document explains how you can use a web browser to test Hotlinks in a deployed voice application.

Prerequisites

Components Used

The information in this document is based on Cisco Unified Call Services, Universal Edition.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Common Questions

How can I test Hotlinks with a web browser?

Symptoms: The developer tests the application in a web browser, such as Internet Explorer or Firefox, but does not know how to test Hotlinks, which normally rely on an event that is thrown by a voice browser. Since web browsers cannot throw these events, it cannot be immediately apparent how to replicate this behavior for testing purposes.

Resolution: In order to test a Hotlink, send a request with a URL of this format

```
http://IP:PORT/Audium/Server?audium_hotlink=HOTLINK_NAME
```

where IP is the address of your application server, and PORT is the port on which it listens. HOTLINK_NAME is the name of the Hotlink that you want to trigger from within your voice application.

Note: In order to issue this request, you must already have accessed your voice application through a web browser and be familiar with how to navigate through your application with the web browser.

You can simulate any thrown event that is defined in the root document of your application in a similar manner. In order to do so, you must examine the root document, and look for the <submit> that exists within the desired <catch> block of the event. An example is a thrown hang-up event. This VoiceXML can be found

in the root document of your applications:

```
<catch event="telephone.disconnect.hangup">  
<var name="audium_action" expr="'hangup'"/>  
<submit next="/Audium/Server" method="post" namelist=" audium_vxmlLog audium_action"/> </c
```

From this VoiceXML, you can construct a web browser request that simulates this event.

Note: The required `audium_action` variable is passed, as seen in the submit. The `audium_vxmlLog` variable is always optional and has been omitted:

```
http://IP:PORT/Audium/Server?audium_action=hangup
```

When Cisco Unified Call Services receives this request, it assumes that a hang-up event has occurred on the voice browser side and acts accordingly.

Note: You can view the root document of your application when you request this URL

```
http://IP:PORT/Audium/Server?audium_root=true&calling_into=YOUR_APP_NAME
```

where `YOUR_APP_NAME` is the name of your voice application.

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for Voice
Service Providers: Voice over IP
Voice & Video: Voice over IP
Voice & Video: IP Telephony
Voice & Video: IP Phone Services for End Users
Voice & Video: Unified Communications
Voice & Video: IP Phone Services for Developers
Voice & Video: General

Related Information

- **Technical Support & Documentation – Cisco Systems**

All contents are Copyright © 2006–2007 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Jul 03, 2007

Document ID: 71485
