

Notify URI

XML Service Interface Application Note



Target Audience: Unified Communication XML Application developers

Concept: This document was created by the Cisco Voice Technology Group as part of a series of documents to provide information and implementation guidance on new XML application interface objects as they become available.

The XML Service Interface documentation and SDK are updated as new versions of Cisco Unified Communication Manager are released. In between releases of Cisco Unified Communication Manager, new releases of IP phone firmware are also released. Some of these IP phone firmware releases may contain enhancements to the XML Service Interface. This series of XML service interface applications notes has been produced as a way of making these enhancements available to the developer community as early as possible.

The information contained in this application note will be incorporated into the next release of the XML application interface documentation and the SDK. At that time, the information in this document will be deprecated, but until that time this document acts as the reference documentation for the XML Service Interface API being described.



Notify URI

Table of Contents

Introduction	2
Description of Operation	2
Syntax	3
Sample Usage	4
Pre-requisites	6
Supported Platforms	6
Error and Response.....	6
Debug / Logging.....	6
Caveats	7
Additional References	7

Introduction

The purpose of this API is to allow XSI action handlers to generate network notifications to backend applications. For example, to deliver notifications to backend applications when an XSI application is closed or when an RTP stream is terminated.

This feature is most useful in the XSI objects which support action handlers (such as displayable XSI objects and RTP streams), but can be specified in place of most fields which accept a generic URI, including soft keys and menu items. This is useful in cases where a Notify URI could be called from a soft key or menu item in order to trigger some backend event which does not necessarily require a change/refresh in the UI – such as manipulating the state of audio streams or other non-visual resources.

Description of Operation

Execution of the Notify URI will trigger a network callback to the specified target, sending the specified data, if present. The Notify URI also supports authentication to the network server via an optional “credentials” parameter. The URI syntax allows for additional protocol support in the future, but currently only HTTP is supported.

For HTTP notifications, the data will be HTTP POSTed using the standard “application/x-www-form-urlencoded” MIME type, with a control name of “DATA”, and the character set (encoding) will always be UTF-8. Abiding by this standard convention will greatly simplify the server-side



processing of the notifications. HTTP Basic authentication is supported and is enabled if the “credentials” parameter is specified and is non-null. For HTTP notifications, the “credentials” parameter must contain a base64-encoded representation of the userid:password, just as it would appear in an HTTP Authorization header.

If the “credentials” parameter is not specified or if it is null (empty), no Authorization header will be included in the request. The HTTP notification service will retry the request 3 times before failing and logging an error message.

NOTE: The Notify URI callback notification will NOT be made in the context of an XSI application session and, therefore, will not contain any HTTP cookie or session information. This means that the backend application CANNOT rely on HTTP cookies or session information to uniquely identify the client/application – the application must embed any necessary information in the Notify path and data fields, or leave the data field empty and rely on the default information provided by the specific event handler, if any.

The Notify URI **will** work in conjunction with the QueryStringParam URI, such that the exact contents of the QueryStringParam data will be used as the Notify URI data.

Syntax

Notify:{protocol}:{host}:{port}:{path}[:{credentials}[:{data}]]

{protocol}

Description: Network protocol to use for the Notify connection

Type: Enumeration

Values/Range: Only 'http' is currently supported

Default-value: N/A

{host}

Description: The Network host to receive the notification

Type: Hostname or IP address

Values/Range: no inherent length limits

Default-value: N/A

{port}

Description: Network port to use for the Notify connection

Type: Numeric

Values/Range: range 1-65535

Default-value: N/A

{path}

Description: Protocol-specific path information

Type: String



Values/Range: no inherent length limits, cannot contain colons or semicolons
Default-value: N/A

{credentials} (optional)

Description: Protocol-specific credentials used to authenticate to the server. For HTTP, this is a base64-encoded version of userid:password

Type: String

Values/Range: no inherent length limits, but cannot contain semicolons or colons

Default-value: null (empty string)

{data} (optional)

Description: Application-specific event data

Type: String

Values/Range: no inherent length limitation, cannot contain semicolons

Default-value: The event data provided by the event handler invoking this URI, if applicable, otherwise null (empty string)

Sample Usage

NOTE: In order to improve readability, the 'DATA' in these sample notifications is **not** shown in its actual URL-encoded format.

Called from RTP onStreamStopped Event Handler, no Credentials, with Data

Notify:http:myserver:8080:path/streamhandler?event=stopped::myStreamStoppedData

HTTP POST /path/streamhandler?event=stopped HTTP/1.1

Accept: */*

Content-Type: application/x-www-form-urlencoded; charset="UTF-8"

Host: myserver:8080

Content-Length: 23

DATA=myStreamStoppedData

Called from RTP onStreamStopped Event Handler, no Credentials, no Data

Notify:http:server:8080:path/streamhandler?event=stopped

HTTP POST /path/streamhandler?event=stopped HTTP/1.1

Accept: */*

Content-Type: application/x-www-form-urlencoded; charset="UTF-8"



Host: myserver:8080
Content-Length: 40

DATA=<notifyStreamStopped id="stream1"/>

Called from SoftKey, with Credentials, with Data

Notify:http:myserver:8080:path/streamhandler?event=stopped:8fh4hf7s7d
hf :myStreamStoppedData

HTTP POST /**path/streamhandler?event=stopped** HTTP/1.1
Accept: */*
Authorization: Basic **8fh4hf7s7dhf**
Content-Type: application/x-www-form-urlencoded; charset="UTF-8"
Host: **myserver:8080**
Content-Length: 23

DATA=**myStreamStoppedData**

Called from SoftKey, no Credentials, no Data

Notify:http:server:8080:path/streamhandler?event=stopped

HTTP POST /path/streamhandler?event=stopped HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset="UTF-8"
Host: myserver:8080
Content-Length: 5

DATA=

Called from SoftKey with QueryStringParam URI

```
<CiscoIPPhoneMenu>
  <MenuItem>
    <Name>Voicemail1</Name>
    <URL>QueryStringParam:id=1</URL>
  </MenuItem>
  <MenuItem>
    <Name>Voicemail2</Name>
    <URL>QueryStringParam:id=2</URL>
  </MenuItem>
  <SoftKeyItem>
    <Name>Play</Name>
    <URL>Notify:http:vmalSrvr:8080:path/play</URL>
  </SoftKeyItem>
</CiscoIPPhoneMenu>
```



If the 'Voicemail2' menu item was selected when the 'Play' soft key was pressed, the following notification would be sent:

```
HTTP POST /path/play HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset="UTF-8"
Host: vmailSrvr:8080
Content-Length: 9
```

```
DATA=id=2
```

Pre-requisites

This XML Service Interface API, requires IP phone firmware 8.3(2) or later. For the availability of the 8.3(2) phone firmware, please consult www.cisco.com or contact the Developer Support team in order to obtain an early release of this phone firmware.

http://www.cisco.com/en/US/products/svcs/ps3034/ps5408/ps5418/serv_home.html

NOTE: Phone Firmware 8.3(2) contains an updated XML parser, with more rigid enforcement of the XML schema. XML applications that have been written to earlier versions of the XML parser, may encounter exceptions if they do not conform to the XML schema. It is recommended that any existing XML applications be validated with the new XML parser before trying to enhance the XML application to use this new XML Service Interface API.

For details on the new XML parser, please consult the XML Service Interface Application note entitled **IP Phone Services XML Schema Enforcement**

Supported Platforms

The following models of IP phones will support this XML Service Interface API, 7906, 7911, 7931, 7941, 7961, 7970, and 7971.

The 7905, 7912, 7920, 7940, 7960 and 7985 will not support this XML Service Interface API

Error and Response

The Notify URI is not supported in the Execute object therefore, no response or error codes are supported.

Debug / Logging

Any error conditions are sent to the logging service at level ERROR.



Caveats

Since there are no common use cases for invoking the Notify URI from an Execute item, the use of Notify URI from an Execute item is not supported.

The Notify URI syntax supports a “credentials” field to allow authentication of notification connections. For this implementation, only HTTP Basic authentication is supported and the “credentials” field contains the base64-encoded userid/password. Support for HTTPS is planned for a later release of IP phone firmware.

Additional References

Cisco IP phone Services along with developer resources such as the SDK to assist developers in getting started with applications and documentation on the various URIs and XML Objects that are supported in the IP phones are available at the link posted below.

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_usage_guide_book09186a00807a34b9.html

In recent years, Cisco has introduced a new generation of desktop IP phone models, the 7906, 7911, 7931, 7941, 7961, 7970 and 7971. These new models of phones have been introduced as a platform in which new features can be introduced, and on which new applications can be built. The introduction of these new generation IP phones has also provided an opportunity to increase the range of XML objects and URIs that are supported. The introduction of enhancements to IP phone services is also the time to remind application developers, of the XML schema that is defined for the IP phone services. This schema is published at the following location, and has been since initial release

http://www.cisco.com/en/US/products/sw/voicesw/ps556/products_programming_usage_guide_chapter09186a00807a352c.html

Please contact the developer support team and gain more information about Developer services and access to early deployment phone loads.

http://www.cisco.com/en/US/products/svcs/ps3034/ps5408/ps5418/serv_home.html