HTTP Requests and Header Settings

HTTP Requests and Header Settings Overview

Cisco Unified IP Phones use HTTP to communicate to external applications. The phone firmware includes an HTTP client for making requests and an HTTP server for receiving requests. This chapter describes the capabilities of the HTTP interface.

HTTP Client Requests (HTTP GET)

The following description designates how HTTP client requests are handled:

1. The phone HTTP client performs an HTTP GET for a specified URL.

2. The HTTP server processes request and returns an XML object or plain text.

3. The phone processes the supported HTTP headers.

4. The phone parses the XML object if *ContentType* is text/xml.

5. The phone presents data and options to the user, or in the case of a *CiscoIPPhoneExecute* object, begins executing the URIs.

HTTP Server Requests (HTTP POST)

The following description designates how an HTTP server request is made to the phone using an HTTP POST operation:

1. The server performs an HTTP POST in response to a case-sensitive URL of the phone with this format: *http://x.x.x.x/CGI/Execute*, where *x.x.x.x* represents the IP address of the destination Cisco Unified IP Phone.

   The form that is posted should have a case-sensitive form field name called "XML" that contains the desired XML object. For any HTTP POST operation, the server must provide basic HTTP authentication information with the POST. The provided credentials must be of a user in the global directory with a device association with the target phone.

   If the credentials are invalid, or the Authentication URL is not set properly in the Cisco Unified Communications Manager Administration, the phone will return a CiscoIPPhoneError with a value of 4 (Authentication Error) and processing will stop.

2. The phone processes the supported HTTP headers

3. The phone parses and validates the XML object

4. The phone presents data and options to the user, or in the case of a *CiscoIPPhoneExecute* object, begins executing the URIs.

Any HTTP POST object is limited to 40 Kbytes in size. Larger objects (such as images) can only be delivered to the phone using HTTP GET. To push large objects to the phone, the server application must take an indirect approach, by pushing an Execute object to the phone that contains an *ExecuteItem* that points to the URL of the large object.

Note JTAPI can push an XML object directly to an IP phone, with the added benefit of not requiring authentication (because the JTAPI connection itself is already authenticated). This option works particularly well for adding XML services interfaces to existing CTI applications (where the overhead of the CTI connection is already a requirement). Objects pushed using JTAPI are also limited to a maximum size of 512 bytes. For more information, see the *Cisco Unified Communications Manager JTAPI Developer Guide*.

HTTP Header Settings

The following list provides definitions for HTTP header elements for Cisco Unified IP Phone Services:

- Refresh: sets the refresh time (in seconds) and URL
    - If no time is set or it is zero, the refresh gets set to manual.
    - If no URL is set, the current URL gets used.

- ContentType: notifies the phone of the MIME type that was sent.
- Expires: sets the Date/Time in GMT when the page is to expire.

  Pages that have expired before being loaded do not get added to the URL stack in the phone. The phone does not cache content.

- Set Cookie
- HTTP encoding header

HTTP Refresh Setting

The HTTP headers that are sent with any page from an HTTP server can include a Refresh setting. This setting comprises two parameters: a time in seconds and a URL. These two parameters direct the recipient to wait the time given in the seconds parameter and then get the data to which the URL points.

The Cisco Unified IP Phone HTTP client properly supports this setting, which gives a great deal of power to service developers. It means that a new page can replace any XML object that displays after a fixed time.

The following figure shows an example of how to use the refresh setting. This sample page shows the user the current value of Cisco stock.

- A splash screen that displays the Yahoo logo.

- After a very short time, the screen displays the numeric Cisco stock parameters.

- Finally, the screen shows a graph of Cisco intraday stock performance. The display repeatedly cycles between the final two views.

Figure 1: Refresh Display Sample
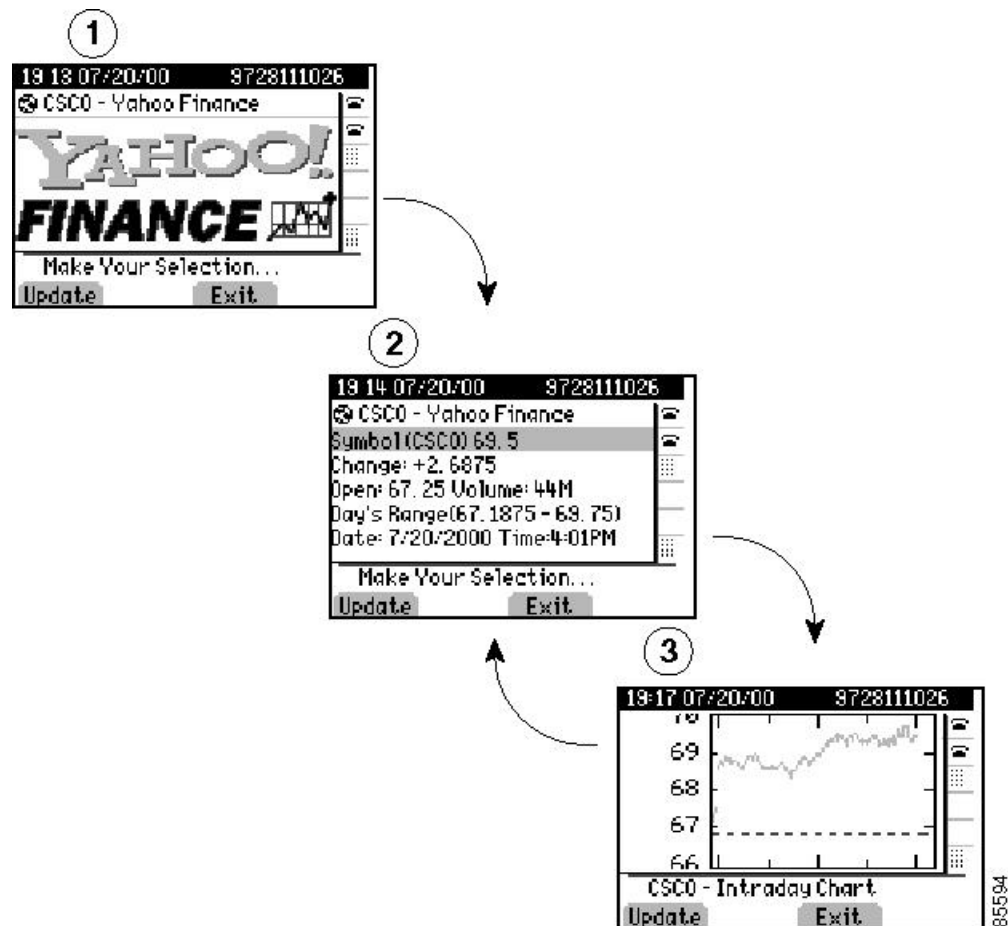
Refreshing the display can occur without user intervention, because the display automatically cycles if a timer parameter is specified. On any given screen, however, the user can force an immediate reload by pressing the Update softkey. Also, if a timer parameter of 0 was sent in the header, the page never automatically reloads.

In this case, the display will move to the next page only when the Update softkey is pressed. If no refresh URL is specified, the current page gets reloaded.

MIME Type and Other HTTP Headers

Although delivering pages with the proper MIME type and other formatting items is not difficult, you require a moderately in-depth knowledge of your web server. The following code excerpt, written in JavaScript and used with Microsoft IIS and ASP, sets these values in a few lines:

```
<%@ Language=JavaScript %>
<%
Response.AddHeader( "Refresh",
                     "3; url=http://services.cisco.com/s/q.asp");
Response.ContentType = "text/xml";
//
// Additional page content here
//
%>
```

Usually, you can set the MIME type for pages in any web server by simply performing an association to the .xml file extension. Your web server documentation should explain how to accomplish this association. This action allows you to serve static pages without the need for writing script.

If you want to deliver dynamic content by using the other supported HTTP headers, you need to understand how to generate the HTTP headers using the desired programming language and have common gateway interface (CGI) or script access on the target web server.

Audio Clips

You can serve audio clips to the phone from a web server by using the "audio/basic" MIME type setting. When this MIME type is used, the body of the response should contain raw audio data in the same format that is used for custom IP phone rings. For more information, see the "Custom Phone Rings" chapter in the *Cisco Unified Communications Manager System Guide* (also available in the online help) and the phone administration guide.

Note The audio file should not be longer than five seconds.

Use the following ASP sample script to set the MIME type and to serve the file that is specified in the #include command:

```
<%@ Language=JavaScript%>
<%
Response.ContentType = "audio/basic";
%><!--#include file="filename.raw" --><% Response.End();%>
```

Using a script to generate the MIME header when playing a sound provides an advantage because you may also include a refresh header to take the phone to a subsequent URL. Usually, you can set the MIME type for pages in any web server by simply performing an association to the .xml or .raw file extension. Your web server documentation should explain how to accomplish this. This action allows you to serve static pages without the need for writing script.

Content Expiration Header Setting

The expiration header can control which URLs are added to the phone URL history. This behavior differs slightly from traditional web browsers but is implemented to perform the same function. Disable the Back button functionality to avoid calling a URL twice.

This functionality allows you to expire the content of any page that is sent to the phone. When a user presses the Exit softkey, the user goes back to the last URL that did not expire when it was loaded. This action differs from traditional browsers by not considering the current freshness of the data but the freshness of the data when the URL was requested. This functionality requires you to have a page expire when it is first loaded and to not set a time and date in the future.

The following example shows how to have content on IIS expire by using Active Server Page (ASP):

```
<%@ Language=JavaScript %>
<%
   Response.ContentType = "text/xml";
   Response.Expires = -1;
%>
```

The "Expires" property specifies the number of minutes to wait for the content to expire. Setting this value to -1 subtracts 1 minute from the request time and returns a date and time that have already passed.

Note The Cisco IP Phone 8800 Series requires that the expire date be after 1970/1/1.

Set-Cookie Header Setting

A *cookie* is the term for a mechanism that the Web server uses to give the client a piece of data and have the client return the data with each request. The two traditional uses for cookies are:

For Web sites to store a unique identifier or other information on the client's file system. The information is available to the Web server on subsequent visits.

To track a unique identifier for state management. The client returns the cookie with each request and the server uses this identifier to index information about the current session. The identifier is commonly referred to as a session ID. Most Web servers have a built-in session management layer that uses this second type of cookie, which is commonly referred to as a session cookie.

The following example shows the Set-Cookie header that is returned to the browser when a request method is used:

```
Set-Cookie: ASPSESSIONIDGQGQGRLS=OCPNMLFDBJIPNIOOKFNFMOAL; path=/
```

The Cisco Unified IP Phone can receive and use a total of four cookies per host per session and can store information for up to eight sessions at once. Each cookie can be up to 255 bytes in size. These cookies are available until the server terminates the session or the client session has been idle for more than 30 minutes. On many phones which are capable of running multiple applications concurrently, the session state is also cleared when the application window closes. This behavior is consistent with PC-based browsers and provides better security because anyone attempting to reopen a secure application would be forced to authenticate. If the client is connecting to a new server and all session resources are in use, the client clears and reuses the session with the longest inactivity time.

When using ASP on IIS, the default server configuration automatically generates a session cookie and sends the cookie to the client using the Set-Cookie header. This cookie enables you to use the Session object from

within ASP to store and retrieve data spanning multiple requests for the life of the session. When using JSP on Tomcat, the default configuration generates and issues a session cookie.

HTTP Encoding Header Setting

The encoding header controls language and character settings related to localization.

Accept Language

Cisco Unified IP Phones populate the Accept-Language HTTP request header in compliance with the HTTP specification.

For example, the Accept-Language value advertised by a phone configured for the English_United_States user locale would look like:

```
Accept-Language: en-US
```

Accept Charset

The phones are capable of handling UTF-8 encoding and, depending on phone model, some degree of Unicode support.

In addition to the character set for the currently configured user locale, the phones also support ISO-8859-1 characters in their font files.

All phones advertise their supported encodings using the standard HTTP Accept-Charset header. According to the HTTP standard, q-values are used to specify preferred encodings. The older phone models, with more limited UTF-8 support, specify a lower q-value for UTF-8 than the default user locale encoding.

For example, an older phone model configured with the English_United_States user locale would include an Accept-Charset header similar to the following:

```
Accept-Charset: iso-8859-1, utf-8;q=0.8
```

A newer phone model with Unicode support would advertise an Accept-Charset similar to the following:

```
Accept-Charset: utf-8,iso-8859-1;q=0.8
```

HTTP Response Headers: Content-Type

Because the phones are capable of supporting multiple character encodings, HTTP responses returned to the phones should include the charset parameter on the HTTP Content-Type header. The following are examples of responses including the charset parameter:

```
Content-Type: text/xml; charset=ISO-8859-1
Content-Type: text/xml; charset=UTF-8
Content-Type: text/plain; charset=Shift_JIS
```

HTTP standards state that if the encoding is not explicitly specified, ISO-8859-1 is the default. Cisco Unified IP Phones are typically compatible with this spec, but not fully compliant.

If charset is not specified, the phones use the default encoding for the currently configured user locale. To avoid possible problems where the default encoding may not be ISO-8859-1, the web server should explicitly set the Content-Type charset to match one of the Accept-Charset values specified by the phone.

IP Phone Client Capability Identification

XML services are supported on many Cisco IP Phones, so web application servers must identify the capabilities of the requesting IP phone to optimize the content returned to the phone. For example, if the requesting phone is a Cisco IP Phone 7841, which cannot support color PNG images, the application server must be able to identify this limitation and return a gray scale CIP image instead.

The IP phone client request to send the relevant information from the IP phone to the web server application includes these HTTP headers:

- x-CiscoIPPhoneModelName

- x-CiscoIPPhoneDisplay

- x-CiscoIPPhoneSDKVersion

x-CiscoIPPhoneModelName

This Cisco-proprietary header contains the Cisco manufacturing Model Name of the device, which can typically be found by going to the Settings menu on the phone. The navigation path vary from models. Here are some examples:

- On Cisco IP Phones with Multiplatform Firmware:

  Settings > **Status** > **Product**

- On Cisco IP Phones On-Premises:

  Settings > **System Information** > **Phone Information**

  On Cisco Video Phone 8875 and Cisco Desk Phone 9800 Series:

  Settings > **About this device**

Some examples of manufacturing Model Names are CP-7861, CP-8875, CP-8845, and DP-9841.

x-CiscoIPPhoneDisplay

This Cisco-proprietary header contains the display capabilities of the requesting device with the following parameters (listed in the order in which they appear):

- Width (in pixels)

- Height (in pixels)

- Color depth (in bits)

- A single character indicating whether the display is color ("C") or gray scale ("G")

These parameters get separated by commas as shown in the following example header:

```
x-CiscoIPPhoneDisplay: 298, 168, 12, C
```

Note The pixel resolutions advertised by the device define the area of the display accessible by the phone services; not the actual resolution of the display.

x-CiscoIPPhoneSDKVersion

This Cisco-proprietary header contains the version of the IP Phone Services SDK that the requesting phone supports. The HTTP header does not specify which URIs are supported. Therefore, you must check the Supported URIs matrix in the IP Phone Services SDK to determine which URIs are supported based on the Phone Model Name and supported SDK version.

Note Beginning with the IP Phone Services SDK 3.3(3), the SDK version number matches the minimum Cisco Unified Communications Manager software that is required to support it. For example, SDK version 3.3(4) gets supported only on Cisco Unified Communications Manager version 3.3(4) or later.

Related Topics

Supported URIs by Phone Model

Accept Header

The Accept header represents a standard HTTP header that is used to inform web servers about the content-handling capabilities of the client.

Cisco Unified IP Phones include proprietary content-types to indicate which XML objects are supported. These proprietary content-types all begin with x-CiscoIPPhone, to indicate Cisco Unified IP Phone XML objects, followed by a slash "/", followed by either a specific XML object or a "*" to indicate all objects.

For example, x-CiscoIPPhone/* indicates that all XML objects defined in the specified version of the SDK are supported, and x-CiscoIPPhone/Menu specifies that the **<CiscoIPPhoneMenu>** object gets supported.

As the Menu example illustrates, the name of the XML object can be derived directly from the content-type by appending the sub-type (the part after the slash) onto CiscoIPPhone. The content-type can also include an optional version to indicate support for a particular SDK version of that object. If a version is not specified, then the x-CiscoIPPhoneSDKVersion is implied. The syntax of the version number may vary, but, in general, will be as follows:

```
<major version>.<minor version>.<maintenance version>
```

Here are some examples of typical content-types:

- x-CiscoIPPhone/*;version=3.3.3

- x-CiscoIPPhone/Text

- x-CiscoIPPhone/Menu;version=3.3.4

IP Phone Information Access

Cisco Unified IP Phones have an embedded web server to provide a programming interface for external applications, and a debugging and management interface for system administrators.

You can access the administrative pages using a standard web browser and pointing to the IP address of the phone with: **/http://<phoneIP>/**, where **phoneIP** is the IP address of the specific phone.

These device information pages are available in either HTML format for manual debugging purposes, or in XML format for automation purposes. The following table lists the available URLs and their purpose.

Table 1: Device Information URLs

HTML URL	XML URL	Description
/DeviceInformation	/DeviceInformationX	General device information
/NetworkConfiguration	/NetworkConfigurationX	Network configuration information
/EthernetInformation	/EthernetInformationX	Ethernet counters
/PortInformation?n	/PortInformationX?n	Detailed port information, where *n* is a model-specific ethernet port identifier, typically in the range 1-3.
/DeviceLog?n	/DeviceLogX?n	Device logging, debug, and error messages, where *n* is a model-specific log number, typically in the range 0 - 2.
/StreamingStatistics?n	/StreamingStatisticsX?n	Current RTP streaming stats, where *n* is model-specific RTP stream identifier, typically in the range 1-3.
/CGI/Execute (password-protected CGI script)		The target URL of a phone push (HTTP POST) request.
/CGI/Screenshot (password-protected CGI script)		Returns an exact snapshot of the current phone display. The size and format of the image returned is model-specific