Cisco IP Phone Services
Application Development Notes

December 28, 2000

Note The eXtensible Markup Language (XML) Application Programming Interface (API) described in this document is available for developers and Cisco CallManager administrators who want to create applications for Cisco IP Phone services. Developers using this guide are encouraged to join the Cisco Developer Support Program due to the fact that standard Cisco TAC support is limited to the Cisco AVVID installation/configuration and Cisco-developed applications. This new program will provide you a consistent level of support that you can depend on while leveraging Cisco interfaces in your development projects. For more information about the availability date of this program and how to join, please contact us at developer-support@cisco.com.

This document is available on CCO at `http://www.cisco.com/univercd/cc/td/doc/product/voice/sw_ap_to/` under the category "User Application Development Guides."

OL-0921-02

Contents

This document covers the following topics:

Introduction

Cisco IP Phones, such as the Cisco IP Phone 7960 and 7940, can be used to deploy customized client services with which users can interact via the keypad and display. Services are deployed using the HTTP protocol from standard web servers, such as Microsoft's IIS. This document provides the information needed for XML programmers and system administrators to develop and deploy new services.

Note Currently, only Cisco IP Phone 7960 and 7940 models support Cisco IP Phone services. Throughout this document, the term "Cisco IP Phone" applies only to the Cisco IP Phone 7960 and 7940.

Cisco IP Phones have buttons labeled **services** and **directories**. When a user presses the **services** button, a menu of services that have been configured for the phone displays. The user then selects a service from the listing, and the phone's display updates.

The following list gives typical services that might be supplied to a phone:

- Weather
- Stock information
- Contact information
- Company news
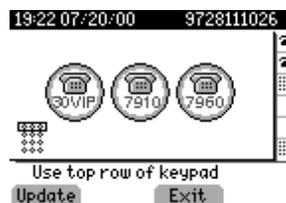- To-do lists
- Daily schedule

Figure 1 shows a sample text menu created during development.

Figure 1   Sample Cisco IP Phone Text Menu

Cisco IP Phones can also display graphic menus, as shown in Figure 2.

Figure 2   Sample Graphic Menu on a Cisco IP Phone

The way a service is configured determines whether a graphic or text menu is displayed.

A phone user can navigate a text menu using the up/down rocker switch followed by the Select soft key, or by using the DTMF keypad to enter a selection directly. Graphic menus do not currently support any type of cursor-based navigation; users simply enter a numeric item selection using the DTMF keypad.

Once a menu selection is made, the Cisco IP Phone acts on it by using its HTTP client to load a specific URL. The return type from this URL can be plain text or one of the `CiscoIPPhone` XML data types. The Cisco IP Phone loads the object, and then interacts with the user in an appropriate manner for the data type.

Figure 3 and Figure 4 show typical displays that result from selecting a service. Figure 3 shows a stock quote generated using plain text.

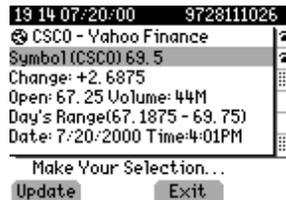*Figure 3   Plan Text Display Example*

Figure 4 displays a graphic image.

Figure 4   Graphic Image Display Example

Cisco IP Phone service activity is limited to a specific pane in the Cisco IP Phone display. A service cannot modify the top line of the phone's display, which contains the time, date, and primary extension. A service cannot overwrite the bottom line of the display, which contains soft key definitions. The pane that displays the service is flush with the left side of the display, and enough of the right side of the display is left intact to ensure that users can see the status of their phone lines.

> **Note** **HTML Disclaimer**: Phone service developers must remember that the phone is *not* a web browser and cannot parse HTML. Although content is delivered to the phone using a web server, the content is not HTML. All content is either sent as plain text or packaged in proprietary XML wrappers.

Cisco IP Phone XML Data Type Definitions

Creating interactive service applications is relatively easy once you master the XML data types that are defined for Cisco IP Phones and the behavior that each data type generates.

The important thing to understand about services is that the phone does not have any concept of a state when it loads an XML page. Cisco IP Phones can use HTTP to load a page of content in many different places, starting with the pressing of the **services** button. Regardless of what causes the phone to load a page, the phone always behaves appropriately once it loads a page.

What is the appropriate behavior? That depends solely on the type of data that has been delivered in the page. This section of the document discusses the six supported XML display types and how they work with Cisco IP Phones.

The web server must deliver the XML pages with a MIME type of text/xml. The exact mechanism required varies according to the type of web server you are using and the server side mechanism you are using to create your pages; for example, static files, JavaScript, CGI, and so on.

Refer to the "MIME Type and Other HTTP Headers" section on page 21 for more information.

The following sections provide definitions and descriptions of each Cisco IP Phone XML data type: `CiscoIPPhoneMenu`, `CiscoIPPhoneText`, `CiscoIPPhoneInput`, `CiscoIPPhoneDirectory`, `CiscoIPPhoneImage`, and `CiscoIPPhoneGraphic`.

CiscoIPPhoneMenu

A menu on the phone comprises a list of text items, one per line. Users select individual menu items using the exact same mechanisms used for built-in menus in the phone, such as those seen in the Cisco IP Phone **settings** pages.

Definition

```
<CiscoIPPhoneMenu>
  <Title>Title text goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <MenuItem>
    <Name>The name of each menu item</Name>
    <URL>The URL associated with the menu item</URL>
  </MenuItem>
</CiscoIPPhoneMenu>
```

The XML format allows you to specify a title and prompt that are used for the entire menu, followed by a sequence of MenuItem objects. Cisco IP Phones allow a maximum of 32 MenuItems. Each MenuItem has a Name and an associated URL.

The behavior of the phone when a menu is loaded is exactly the same as for built-in phone menus. The user navigates through the list of menu items and eventually selects one using either the Select soft key or the DTMF keys.

Once the user selects a menu option, the phone generates an HTTP request for the page with the URL associated with the menu item.

CiscoIPPhoneText

The CiscoIPPhoneText XML data type is used to display ordinary 7-bit ASCII text on the phone display. The text message must not contain any control characters, except for carriage returns, line feeds, and tabs. The Cisco IP Phone firmware controls all other pagination and word-wrap issues.

Definition

```
<CiscoIPPhoneText>
  <Title>Title text goes here</Title>
  <Text>The text to be displayed as the message body goes here</Text>
  <Prompt>The prompt text goes here</Prompt>
</CiscoIPPhoneText>
```

Two optional fields can appear in the XML message:

- The first optional field, `Title`, defines text that is displayed at the top of the display page. If a `Title` is not specified, the `Name` field of the last selected `MenuItem` displays in the `Title` field.

- The second optional field, `Prompt`, defines text that is displayed at the bottom of the display page. If a `Prompt` is not specified, the prompt area of the display pane will be cleared.

Many of the XML objects described in this document also have `Title` and `Prompt` fields. The behavior of these fields normally is identical to that described in this section.

The display shown in Figure 3 provides an example of a text display. The title appears at the top above the text.

Note **Non-XML Text**: This document only describes the six supported Cisco IP XML data types. You can also deliver plain text via HTTP. Pages delivered as MIME type text/html behave exactly the same as XML pages of type `CiscoIPPhoneText`. The only important difference is that you cannot include a title or prompt.

Note **Keypad navigation**: Cisco IP Phones allow navigation to a specific line in a menu by pressing numeric DTMF keys. When a menu is on the display, the actual number to be pressed displays on the left side of the screen.

When normal text displays, the numbers are not shown on the left side of the screen, but the navigation capability is still present. So, a carefully written text service display can take advantage of this capability.

CiscoIPPhoneInput

When a Cisco IP Phone receives an XML object of type CiscoIPPhoneInput, it constructs an input form and displays it. The user then enters data into each of the input items and sends the parameters to the target URL. Figure 5 shows a sample display that is receiving input from a user.

Figure 5   Sample User Input Display

Definition

```
<CiscoIPPhoneInput>
  <Title>Directory title goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <URL>The target URL for the completed input goes here</URL>
  <InputItem>
    <DisplayName>Name of the input field to display</DisplayName>
    <QueryStringParam>The URL query parameter</QueryStringParam>
    <InputFlags></InputFlags>
    <DefaultValue>Value</DefaultValue>
  </InputItem>
 </CiscoIPPhoneInput>
```

The Title and Prompt tags in the object delimit text that is used in the same way as the identical fields in the other Cisco IP Phone XML objects.

The URL tag delimits the URL to which the input results are sent. The actual HTTP request sent to this server is the URL with a list of parameters appended to it as a query string. The parameters are Name/Value pairs, one for each input item.

Note Cisco IP Phones do not use the HTTP POST method.

The InputItem tag is used to delimit each item in the list. The number of InputItems must not exceed five. Each input item has a DisplayName, which is the prompt written to the display for that particular item. Each item also has a QueryStringParam, which is the name of the parameter appended to the URL when it is sent out after input is complete. Each input item can also use the DefaultValue tag to set the default value to be displayed.

The final item for each input item is a set of InputFlags. The following table describes the input types that are currently defined.

InputFlag	Description
A	Plain ASCII text. The DTMF keypad is used to enter text consisting of uppercase and lowercase letters, numbers, and special characters.
T	Telephone number. DTMF digits are the only acceptable input for this field. The acceptable input includes numbers, #, and *.
N	Numeric. Numbers are the only acceptable input.
E	Equation. The acceptable input includes numbers and special math symbols.
U	Uppercase. Uppercase letters are the only acceptable input.
L	Lowercase. Lowercase letters are the only acceptable input.
P	Password field. Individual characters display as they are keyed in using the standard keypad-repeat entry mode. As soon as each character is accepted, it is converted to an asterisk, allowing for privacy of the entered value.
	Note P is the only InputFlag that works as a modifier. For example, specify a value of "AP" in the InputFlag field to use plain ASCII as the input type and to mask the input as a password using an asterisk (*).

Soft Keys

During text entry, Cisco IP Phones display soft keys to assist users with text entry. At this time, the following soft keys are used:

Submit	This indicates that the form is complete, and the resulting URL should be requested via HTTP.
<<	Backspace within a field.
Cancel	Cancels the current input.

Users can navigate between fields with the vertical scroll button used to navigate menus, and so on.

CiscoIPPhoneDirectory

The `CiscoIPPhoneDirectory` XML object is actually incorporated in the phone to support the Directory operation of Cisco IP Phones. Because the functionality already exists in the phone, it has also been made available to phone service application developers as well. Figure 6 shows how an XML `CiscoIPPhoneDirectory` object displays on the phone.

Figure 6   Sample CiscoIPPhoneDirectory Object Display

Definition

```
<CiscoIPPhoneDirectory>
  <Title>Directory title goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <DirectoryEntry>
    <Name>The name of the directory entry</Name>
    <Telephone>The telephone number for the entry</Telephone>
  </DirectoryEntry>
 </CiscoIPPhoneDirectory>
```

Note For the directory listing, the Cisco IP Phone displays the appropriate soft keys that are needed to dial the numbers listed on the display. The soft keys include the Edit Dial soft key, which allows users to insert access codes or other necessary items before dialing.

The Title and Prompt tags in the XML object have the usual semantics. A single CiscoIPPhoneDirectory object can contain a maximum of 32 DirectoryEntry objects. If more than 32 entries must be returned, use multiple CiscoIPPhoneDirectory objects in subsequent HTTP requests.

Custom Directories

You can use the "URL Directories" enterprise parameter and Cisco IP Phone XML objects to display custom directories.

The Cisco CallManager enterprise parameter "URL Directories" points to a URL that returns a CiscoIPPhoneMenu object that extends the **directories** menu. The request for "URL Directories" must return a valid CiscoIPPhoneMenu object, even if has no DirectoryEntry objects.

The following objects are optional. If used, they must be used in the order they are listed:

- Use the CiscoIPPhoneInput XML object to collect search criteria.

- Use the CiscoIPPhoneText XML object to display status messages or errors.

- Use the CiscoIPPhoneDirectory XML object to return a list of directory entries that can be dialed.

You can omit the `CiscoIPPhoneInput` or `CiscoIPPhoneText` objects. You can display multiple `CiscoIPPhoneDirectory` objects by specifying an HTTP refresh header that points to the URL of the next individual directory object, which the user accesses by pressing the Next softkey on the phone.

CiscoIPPhoneImage

Cisco IP Phones have a bit-mapped display with a 133 x 64-pixel pane that is available to services. Each pixel has four greyscale settings. A value of three (3) displays as black, and a value of zero (0) displays as white.

✎

Note The palette is inverted because the phone uses an LCD display.

The `CiscoIPPhoneImage` XML type lets you use the Cisco IP Phone's display to present graphics to the user. Figure 4 shows an example of how this looks on the current release of the Cisco IP Phone.

Definition

```
<CiscoIPPhoneImage>
  <Title>Image title goes here</Title>
  <LocationX>Position information of graphic</LocationX>
  <LocationY>Position information of graphic</LocationY>
  <Width>Size information for the graphic</Width>
  <Height>Size information for the graphic</Height>
  <Depth>Number of bits per pixel</Depth>
  <Data>Packed Pixel Data</Data>
  <Prompt>Prompt text goes here</Prompt>
</CiscoIPPhoneImage>
```

The `CiscoIPPhoneImage` data type definition includes two familiar elements: `Title` and `Prompt`. These elements serve the same purpose as they do in the other Cisco IP Phone XML objects. The `Title` displays at the top of the page, and the `Prompt` displays at the bottom.

Use `LocationX` and `LocationY` to position the graphic on the phone's display. The upper, left corner of the graphic is positioned at the pixel defined by these two parameters. Setting the X and Y location values to (0, 0) positions the graphic at the upper, left corner of the display. Setting the X and Y location values to (-1, -1) centers the graphic in the services pane of the phone's display.

Width and Height are self-explanatory. If the values specified do not match up properly with the pixel stream specified in the Data field, the results are unpredictable and generally incorrect.

Depth specifies the number of bits per pixel. Cisco IP Phones only support a value of 2. Future Cisco IP Phones may have other values.

The Data tag delimits a string of hexadecimal digits that contain the packed value of the pixels in the display. In the Cisco IP Phones, each pixel has only four possible values, which means that you can pack four pixels into a single byte. Each byte is represented as pair of hexadecimal digits.

Figure 7 provides an example of the mechanics of pixel packing. Scanning from left to right in the display, the illustration shows the process for packing consecutive pixel values of 1, 3, 2, and 0. First, the pixels are converted to two-bit binary numbers. Then, the binary pairs are re-ordered in sets of four to create a single re-ordered byte, which is represented by two hexadecimal digits.

Figure 7   Packed Pixel Translation Example
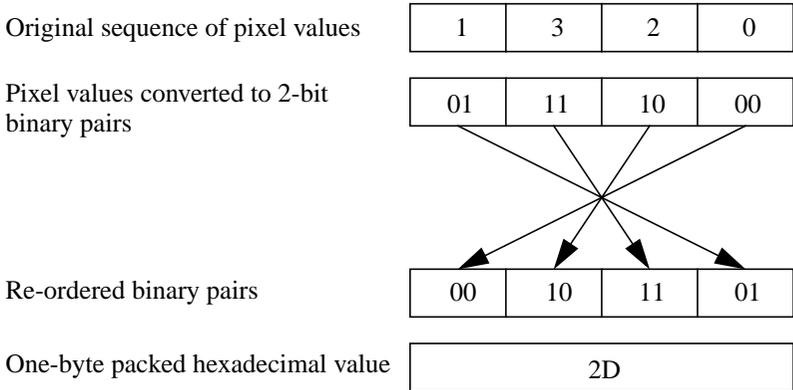
The following XML code defines a `CiscoIPPhoneImage` object that displays the sequence of pixels shown in Figure 7 as a graphic positioned at the center of the phone display:

```
<CiscoIPPhoneImage>
  <Title/>
  <LocationX>-1</LocationX>
  <LocationY>-1</LocationY>
  <Width>4</Width>
  <Height>1</Height>
  <Depth>2</Depth>
  <Data>2D</Data>
  <Prompt/>
</CiscoIPPhoneImage>
```

The graphic display consists of a contiguous stream of hexadecimal digits, with no spaces or other separators. If the number of pixels to be displayed is not an even multiple of four, pad the end of the pixel data with blank (zero value) pixels so that the data is packed correctly. The phone ignores the padded data.

🔍

Tips   When displaying graphics on a Cisco IP Phone, the pane dedicated to services is always cleared before a graphic image is displayed. If a service has text or other information that must be preserved, it must be redrawn as part of the graphic.

This includes the title area. If the title is to be hidden, the graphic must be large enough to cover it.

CiscoIPPhoneGraphicMenu

Graphic menus serve the same purpose as text menus: they allow a user to select a URL from a list. Use graphic menus in situations when the items may not be easy to display in a text list. For example, users might prefer to have their choices presented in a non-ASCII character set such as Kanji or Arabic. Figure 2 shows a graphic menu on a Cisco IP Phone.

In these cases, the information is presented as a bit-mapped graphic. The user then enters a menu selection by using the DTMF keypad to enter a number.

Definition

```
<CiscoIPPhoneGraphicMenu>
  <Title>Menu title goes here</Title>
  <LocationX>Position information of graphic</LocationX>
  <LocationY>Position information of graphic</LocationY>
  <Width>Size information for the graphic</Width>
  <Height>Size information for the graphic</Height>
  <Depth>Number of bits per pixel</Depth>
  <Data>Packed Pixel Data</Data>
  <Prompt>Prompt text goes here</Prompt>
  <MenuItem>
    <Name>The name of each menu item</Name>
    <URL>The URL associated with the menu item</URL>
  </MenuItem>
</CiscoIPPhoneGraphicMenu>
```

Menu items in the graphic menu have a name, just like the text menu counterparts. Although the name is not displayed to the user, it still performs a function. The name of the menu item provides the default title used when the URL for the selected item is loaded. If the loaded page has a title of its own, that will be used instead.

The XML tags in GraphicMenu use the tag definitions for `CiscoIPPhoneImage` and `CiscoIPPhoneMenu`. The semantics of the tags are identical. However, you can only have 12 `MenuItem` objects in a `CiscoIPPhoneGraphic` object. Refer to the "CiscoIPPhoneMenu" section on page 6 and "CiscoIPPhoneImage" section on page 12 for detailed descriptions.

Cisco IP Phone XML Data Type Quick Reference

Table 1   Cisco IP Phone XML Data Type Quick Reference

Data Type	Definition

CiscoIPPhoneMenu	```<CiscoIPPhoneMenu>```
	 `  <Title>Title text goes here</Title>`
	 `  <Prompt>Prompt text goes here</Prompt>`
	 `  <MenuItem>`
	 `    <Name>The name of each menu item</Name>`
	 `    <URL>The URL associated with the menu item</URL>`
	 `  </MenuItem>`
	 `</CiscoIPPhoneMenu>`

CiscoIPPhoneText	`<CiscoIPPhoneText>`
	 `  <Title>Title text goes here</Title>`
	 `  <Text>Text to display as the message body goes here</Text>`
	 `  <Prompt>The prompt text goes here</Prompt>`
	 `</CiscoIPPhoneText>`

CiscoIPPhoneMenu	`<CiscoIPPhoneInput>`
	 `  <Title>Directory title goes here</Title>`
	 `  <Prompt>Prompt text goes here</Prompt>`
	 `  <URL>The target URL for the completed input goes here</URL>`
	 `  <InputItem>`
	 `    <DisplayName>Name of input field to display</DisplayName>`
	 `    <QueryStringParam>The URL query parameter</QueryStringParam>`
	 `    <InputFlags></InputFlags>`
	 `    <DefaultValue>Value</DefaultValue>`
	 `  </InputItem>`
	 ` </CiscoIPPhoneInput>`

CiscoIPPhoneDirectory	`<CiscoIPPhoneDirectory>`
	 `  <Title>Directory title goes here</Title>`
	 `  <Prompt>Prompt text goes here</Prompt>`
	 `  <DirectoryEntry>`
	 `    <Name>The name of the directory entry</Name>`
	 `    <Telephone>The telephone number for the entry</Telephone>`
	 `  </DirectoryEntry>`
	 ` </CiscoIPPhoneDirectory>`

Table 1   Cisco IP Phone XML Data Type Quick Reference

Data Type	Definition
CiscoIPPhoneImage	``` <CiscoIPPhoneImage> <Title>Image title goes here</Title> <LocationX>Position information of graphic</LocationX> <LocationY>Position information of graphic</LocationY> <Width>Size information for the graphic</Width> <Height>Size information for the graphic</Height> <Depth>Number of bits per pixel</Depth> <Data>Packed Pixel Data</Data> <Prompt>Prompt text goes here</Prompt> </CiscoIPPhoneImage> ```
CiscoIPPhoneGraphicMenu	``` <CiscoIPPhoneGraphicMenu> <Title>Menu title goes here</Title> <LocationX>Position information of graphic</LocationX> <LocationY>Position information of graphic</LocationY> <Width>Size information for the graphic</Width> <Height>Size information for the graphic</Height> <Depth>Number of bits per pixel</Depth> <Data>Packed Pixel Data</Data> <Prompt>Prompt text goes here</Prompt> <MenuItem> <Name>The name of each menu item</Name> <URL>The URL associated with the menu item</URL> </MenuItem> </CiscoIPPhoneGraphicMenu> ```

XML Considerations

The XML parser in Cisco IP Phones is not a fully capable XML parser. Do not include any tags other than those defined here in your XML display definitions. The parser does not work with attribute values embedded in tags. For example, the following XML text will not parse:

```
<Title font='arial'>
```

Mandatory Escape Sequences

By XML convention, the XML parser also requires that you escape a few special characters. Table 2 lists characters and their escaped values.

.

Table 2   Escape Sequences for Special Characters

Character	Name	Escape Sequence
&	Ampersand	&
"	Quote	"
'	Apostrophe	'
<	Left angle bracket	<
>	Right angle bracket	>

Escaping text can be tedious, but some authoring tools or scripting languages can automate this task.

HTTP Client Requests

HTTP client requests are handled in the following manner:

- The Cisco IP Phone HTTP client performs an HTTP GET for a specified URL.
- The HTPP server processes request and returns an XML object or plain text.
- The phone processes the supported HTTP headers.
- The phone parses the XML object if ContentType is `text/xml`.
- The phone presents data and options to the user per the server response.

HTTP Header Settings

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

- "Refresh" (Time in Seconds, URL)

    - If no time is set or it is zero, the refresh is set to manual.

    - If no URL is set, the current URL is used.

    Refer to the "HTTP Refresh Setting" section on page 19 for details.

- "ContentType" — The ContentType notifies the phone of the MIME type
  sent. Refer to the "MIME Type and Other HTTP Headers" section on page 21.

- "Expires" — Expires sets the Date/Time in GMT when the page is to expire.

    Pages that have expired before being loaded will not be added to the URL
    stack in the phone. The phone does not cache content. Refer to the "Content
    Expiration Header Setting" section on page 21 for more information.

HTTP Refresh Setting

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

The Cisco IP Phone HTTP client properly supports this setting, which gives a
great deal of power to service developers. It means that any XML object being
displayed can be replaced with a new page after a fixed amount of time.

Figure 8 shows a good example of how the refresh setting can be used. This
sample page shows the user the current value of Cisco's stock. It first brings up a
splash screen that displays the Yahoo logo. After a very short time, it displays the
numeric Cisco stock parameters. Finally, it shows a graph of Cisco's intra-day
stock performance. The display then repeatedly cycles between the final two
views.

Figure 8   Sample Refresh Display

(1) Splash screen

```
19 13 07/20/00    9728111026
⊗ CSCO - Yahoo Finance
        YAHOO!
   FINANCE
   Make Your Selection...
 Update        Exit
```

(2) Current Statistics

```
19 14 07/20/00    9728111026
⊗ CSCO - Yahoo Finance
Symbol (CSCO) 69. 5
Change: +2. 6875
Open: 67. 25 Volume: 44M
Day's Range(67. 1875 - 69. 75)
Date: 7/20/2000 Time:4:01PM
   Make Your Selection...
 Update        Exit
```

(3) Intra-day Performance Chart

```
19:17 07/20/00    9728111026
    70
    69
    68
    67
    66
   CSCO - Intraday Chart
 Update        Exit
```

All of this can occur without user intervention, as the display automatically cycles if a timer parameter is specified. However, on any given screen, the user can force an immediate reload by pressing the Update soft key. Also, if a timer parameter of 0 was sent in the header, the page will never automatically reload. In that case, the display will only move to the next page when the Update soft key is pressed. If no refresh URL is specified, the current page will be reloaded.

MIME Type and Other HTTP Headers

Delivering pages with the proper MIME type and other formatting items is not difficult, but it requires moderately in-depth knowledge of your web server. The following code excerpt, written in JavaScript and used with Microsoft's 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's documentation should explain how to accomplish this. This allows you to serve static pages without the need for writing script.

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

Content Expiration Header Setting

The expiration header can control what URLs are added to the phone's URL history. This is slightly different behavior than 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 soft key, they are taken back to the last URL that was not expired when it was loaded. This 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 requires you to expire a page when it is first loaded and to not set a time and date in the future.

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

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

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

IP Phone Service Administration and Subscription

Cisco CallManager administrators have responsibility for maintaining the list of services to which users can subscribe. Add and administer Cisco IP Phone services through the Cisco CallManager Administration web pages:

- To access phone service administration, open Cisco CallManager Administration and select **Feature** > **Cisco IP Phone Services**.

- Phone services can have any number of parameters associated with them.

- Phone service parameters can be optional or required, depending on how they are defined by the phone service application.

- Users can subscribe to any service configured in their cluster.

- Service subscriptions are currently on a device basis.

The core of each service is a URL. When a service is selected from the menu, the URL is requested via HTTP, and a server somewhere provides the content. This URL is entered in the Service URL field. For the services to be available, the phones in the Cisco CallManager cluster must have network connectivity to the server.

For example:

```
http://<servername>/ccmuser/sample/sample.asp
```

Where *<servername>* is a fully qualified domain name or an IP address.

Adding a Phone Service

Figure 9 shows the page the administrator uses to add new services to the system.

- To access this page in Cisco CallManager Administration, select **Feature** > **Cisco IP Phone Services**.

- You can insert, update, or delete a service definition.

- After a service is inserted, you can insert, update, or delete service parameter definitions.

- When you delete a Cisco IP Phone subscription, all service information, user subscriptions, and user subscription data are removed from the database.

- The **Update Subscriptions** button rebuilds all of the user subscriptions if the service has been modified after subscriptions exist.

⚠
Caution   Do not put Cisco IP Phone services on any Cisco CallManager server at your site or any server associated with Cisco CallManager, such as the TFTP server or directory database publisher server. This precaution eliminates the possibility of errors in a Cisco IP Phone Service application having an impact on Cisco CallManager performance or interrupting call processing services.

Figure 9   Adding a New Service in Cisco CallManager Administration
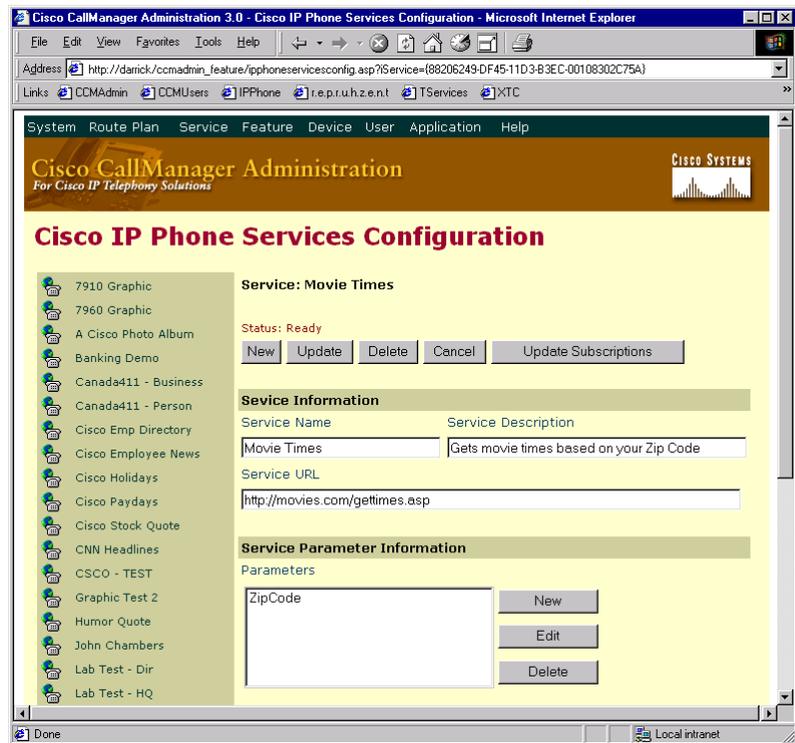
Defining IP Phone Service Parameters in Cisco CallManager Administration

Each service can have a list of parameters. These parameters, which are appended to the URL when sent to the server, can be used to personalize a service for an individual user. Examples of parameters are stock ticker symbols, city names, or user IDs. The service provider entirely defines the semantics of a parameter. The Cisco IP Phone Service Parameter Configuration screen in Cisco CallManager Administration contains the following fields:

Field	Description
Parameter Name	This field provides the exact query string parameter to use when building the subscription URL.
Parameter Display Name	This field designates the descriptive parameter name to display to the user on the user preferences page.
Default Value	This field designates the default value for the parameter. This value displays to the user when a service is being subscribed to for the first time.
Parameter is Required	Check the Parameter is Required box if the user must enter data for this parameter before the subscription can be saved.
Parameter Description	While subscribing to the service, the user can click on a link to view the text entered here. The parameter description should provide information or examples to help users input the correct value for the parameter.

Tips   If you change the service URL, remove a Cisco IP Phone service parameter, or change the Parameter Name of a phone service parameter for a phone service to which users are already subscribed, be sure to click **Update Subscriptions** to update all currently subscribed users with the changes. If you do not do so, users must resubscribe to the service to rebuild the URL correctly.

Figure 10 displays an example Cisco IP Phone Service Parameter Configuration screen in Cisco CallManager Administration.

Figure 10 Defining Service Parameters in Cisco CallManager Administration
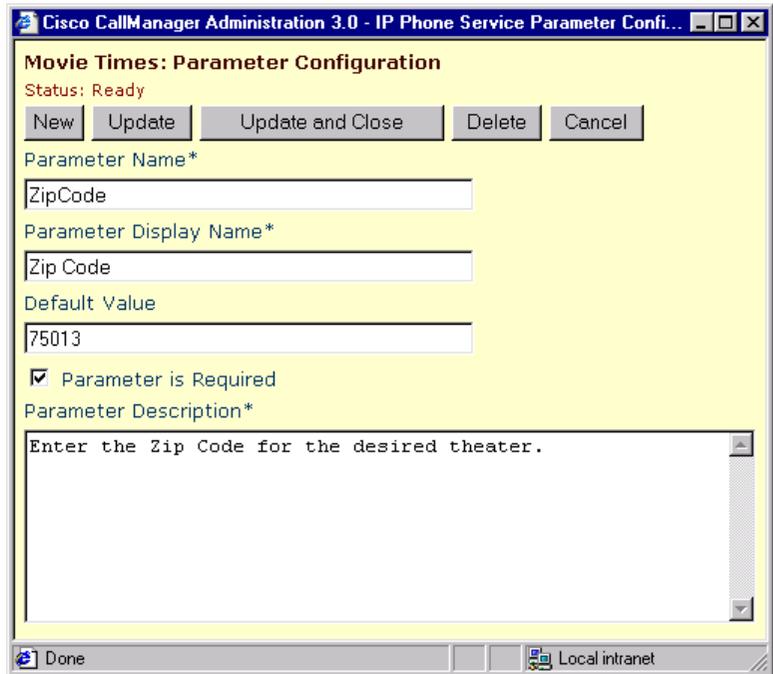
User Service Subscription

User service subscriptions are configured with the CCMUser web site after the user has logged in and selected a device.

The end user has a single list of services that are attached to an individual phone. The user configures this list of services via the user pages, which are also used to set speed dial numbers and call forwarding options. These password-protected pages are authenticated via the LDAP directory.

Figure 11 and Figure 12 show user subscription screens for administering personal services. Users can insert or remove services.

These user configuration pages are the point where values are assigned to the service parameters, allowing for personalization:

- The user can customize the name of the service when it displays on the services list.
- The user can enter any service parameters available for the selected service.
- The user can review the description of each parameter.
- After all the required fields are set, the user can click Subscribe.
- A custom URL is built and stored in the database for this subscription.
- The service then appears on the device's services list.

Figure 11   Cisco CallManager User Subscription Screen
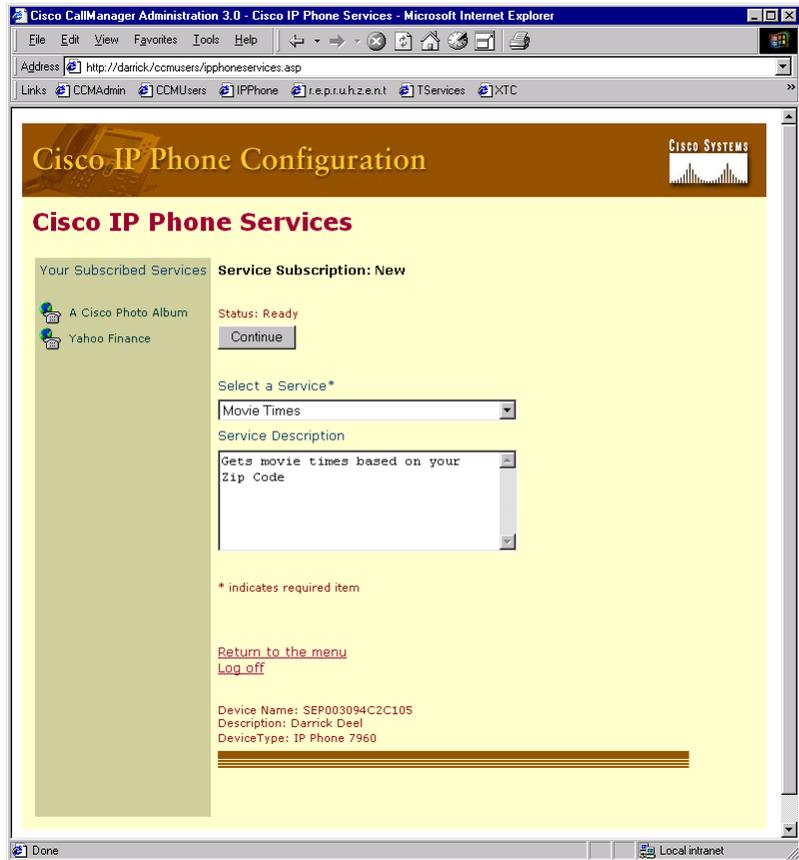
Figure 12   Personalizing a User Subscription
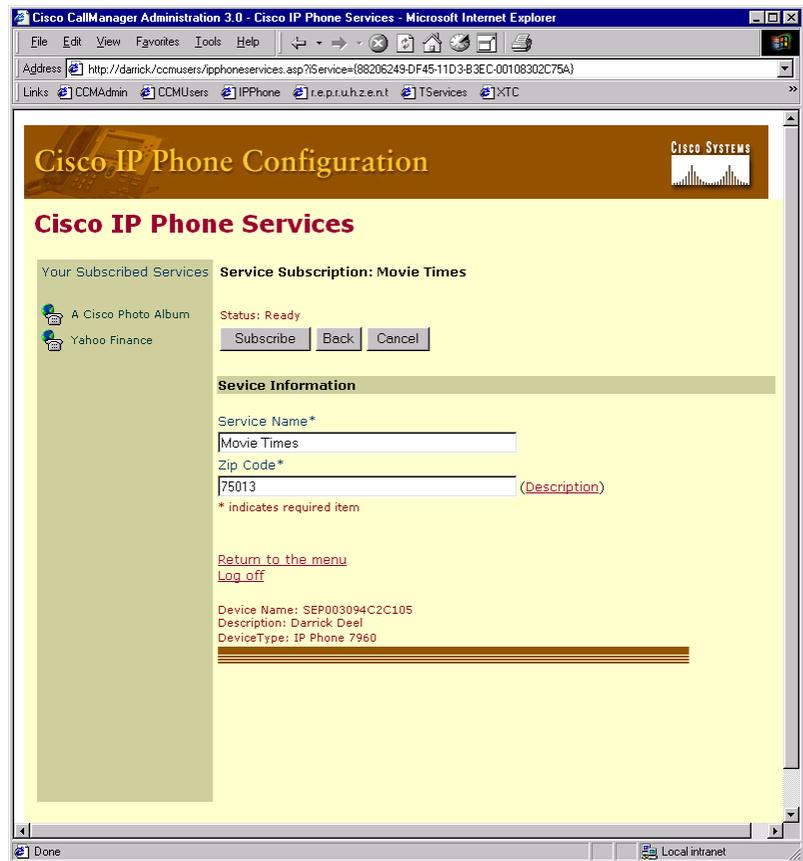
Troubleshooting

The following tips apply to troubleshooting Cisco IP Phone service applications:

- Microsoft Internet Explorer 5 or higher can display the XML source with its default style sheet.

- Standard IP troubleshooting techniques are important for HTTP Errors.

- Externally verify name resolution (Phone has DNS set).

- If DNS is suspected, use IP addresses in URLs.

- Browse the URL in question with IE5 or download and verify with Netscape.

Use a logged telnet session to verify that the desired HTTP headers are returned (Telnet to the server on port 80, then enter get /path/page).

XML Parsing Errors

The following tips apply to troubleshooting XML parsing errors in Cisco IP Phone Services applications:

- Verify the object tags (the object tags are case-sensitive).

- Verify that "&" and the other four special characters are used per the restrictions while inside the XML objects. Refer to "Mandatory Escape Sequences" section on page 18.

Error Messages

Error messages appear on the prompt line of the Cisco IP Phone's display. Only two messages should appear:

- `XML Error[4] = XML Parser error (Invalid Object)`
- `HTTP Error[10] = HTTP Connection Failed`

Related Documentation

The following documents provide further information:

- *Cisco CallManager Administration Guide* (also available in the online help). Refer to the chapter on configuring Cisco IP Phone services.

- *Cisco IP Phone 7960/7940 Quick Start Guide* provides instructions for users on subscribing to phone services.