



# CiscoIPPhone XML Objects

---

- [Object Behavior, on page 1](#)
- [XML Object Support, on page 1](#)
- [XML Object Definitions, on page 6](#)
- [Custom Softkeys, on page 28](#)
- [XML Considerations , on page 29](#)
- [Application Event Handlers, on page 30](#)

## Object Behavior

You can create interactive service applications when you understand the XML objects that are defined for Cisco IP Phones and the behavior that each object generates.

When a phone loads an XML page, the phone does not have any concept of a service state. IP phones can use HTTP to load a page of content in many different places, starting when the user presses the Services button. Regardless of what causes the phone to load a page, the phone always behaves appropriately after it loads a page.

Appropriate behavior depends solely on the type of data that has been delivered in the page. The web server must deliver the XML pages with a MIME type of text/xml. However, the exact mechanism required varies according to the type of web server that you use and the server-side mechanism that you use to create your pages (for example, if you use static files, JavaScript, or CGI).

### Related Topics

[HTTP Requests and Header Settings](#)

## XML Object Support

The following sections describe the supported XML objects by phone model families. Before creating a service for a particular phone model, check to make sure that the XML object you want to use is supported.

## Cisco IP Phone 7800 Series XML Object Support

This section applies to the phones when controlled by Cisco Unified Communications Manager. For information on the phones when controlled by third-party call control systems, see [Multiplatform Phone Support for XML Applications and Services, on page 5](#).

The following table shows the supported XML objects for the Cisco IP Phone 7800 Series and the Cisco IP Conference Phone 7832.

**Table 1: XML Objects Supported by Cisco Unified IP Phone Services SDK for Cisco IP Phone 7800 Series and Cisco IP Conference Phone 7832**

XML object	7811, 7821, 7841, and 7861	7832
CiscoIPPhoneMenu	Supported	Supported
CiscoIPPhoneText	Supported	Supported
CiscoIPPhoneInput	Supported	Supported
CiscoIPPhoneDirectory	Supported	Supported
CiscoIPPhoneImage	Not supported	Not supported
CiscoIPPhoneImageFile	Not supported	Not supported
CiscoIPPhoneGraphicMenu	Not supported	Not supported
CiscoIPPhoneGraphicFileMenu	Not supported	Not supported
CiscoIPPhoneIconMenu	Supported	Supported
CiscoIPPhoneIconFileMenu	Supported	Supported
CiscoIPPhoneStatus	Not supported	Not supported
CiscoIPPhoneStatusFile	Not supported	Not supported
CiscoIPPhoneExecute	Supported	Supported
CiscoIPPhoneResponse	Supported	Supported
CiscoIPPhoneError	Supported	Supported

## Cisco IP Phone 8800 Series XML Object Support

This section applies to the phones when controlled by Cisco Unified Communications Manager. For information on the phones when controlled by third-party call control systems, see [Multiplatform Phone Support for XML Applications and Services, on page 5](#).

The following table shows the supported XML objects for the Cisco IP Phone 8800 Series and Cisco IP Conference Phones 8830 Series.

Table 2: XML Objects Supported by Cisco Unified IP Phone Services SDK for Cisco IP Phone 8800 Series and Cisco IP Conference Phone 8830 Series

XML object	8821	8831	8832	8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR (see note 1)	8875, 8875NR
CiscoIPPhoneMenu	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneText	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneInput	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneDirectory	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneImage	Supported	Not supported	Supported	Supported	Supported
CiscoIPPhoneImageFile	Supported	Not supported	Supported	Supported	Supported
CiscoIPPhoneGraphicMenu	Supported	Not supported	Supported	Supported	Not supported
CiscoIPPhoneGraphicFileMenu	Not supported	Not supported	Supported	Supported	Not supported
CiscoIPPhoneIconMenu	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneIconFileMenu	Supported	Not supported	Supported	Supported	Supported
CiscoIPPhoneStatus	Not supported	Not supported	Supported	Supported	Not supported
CiscoIPPhoneStatusFile	Not supported	Not supported	Supported	Supported	Not supported
CiscoIPPhoneExecute	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneResponse	Supported	Supported	Supported	Supported	Supported
CiscoIPPhoneError	Supported	Supported	Supported	Supported	Supported



**Note** Check the required phone hardware and firmware versions before you configure XSI services for your phones:

- Cisco IP Phone 8811 requires Firmware Release 10.2(2) or later.
- Cisco IP Phone 8851NR requires Firmware Release 10.3(1) or later.
- Cisco IP Phone 8845 and 8865 require Firmware Release 10.3(2) or later.
- Cisco IP Phone 8865NR requires Firmware Release 11.5(1) or later.
- Cisco IP Phone 8811, 8845, 8851, and 8851NR (hardware version 08 or later) require Firmware Release 11.7(1) or later.
- Cisco Video Phone 8875 and 8875NR require PhoneOS 2.1 or later. More XML objects will be supported in the future releases of PhoneOS.

## Cisco Desk Phone 9800 Series XML Object Support

This section applies to the phones that are registered with Cisco Unified Communications Manager. The currently release doesn't support XML applications on phones that are registered with Cisco BroadWorks.

**Table 3: XML Objects Supported by Cisco Desk Phone 9800 Series**

XML object	9841	9851
CiscoIPPhoneMenu	Supported	Supported
CiscoIPPhoneText	Supported	Supported
CiscoIPPhoneInput	Supported	Supported
CiscoIPPhoneDirectory	Supported	Supported
CiscoIPPhoneImage	Supported	Supported
CiscoIPPhoneImageFile	Supported	Supported
CiscoIPPhoneGraphicMenu	Not supported	Not supported
CiscoIPPhoneGraphicFileMenu	Not supported	Not supported
CiscoIPPhoneIconMenu	Supported	Supported
CiscoIPPhoneIconFileMenu	Supported	Supported
CiscoIPPhoneStatus	Not supported	Not supported
CiscoIPPhoneStatusFile	Not supported	Not supported
CiscoIPPhoneExecute	Supported	Supported
CiscoIPPhoneResponse	Supported	Supported
CiscoIPPhoneError	Supported	Supported

## Cisco Wireless Phone 800 Series XML Object Support

Cisco Wireless Phone 840 or 860 Release 1.10(0) supports a subset of the XSI API.

The following table shows the supported XML objects for the Cisco Wireless IP Phone 8821 and Cisco Wireless Phone 800 Series.

**Table 4: XML Objects Supported by Cisco Unified IP Phone Services SDK for Cisco IP Phone 8821 and Cisco Wireless Phone 800 Series**

XML object	8821	840/860
CiscoIPPhoneMenu	Supported	Supported
CiscoIPPhoneText	Supported	Supported
CiscoIPPhoneInput	Supported	Not Supported

XML object	8821	840/860
CiscoIPPhoneDirectory	Supported	Not Supported
CiscoIPPhoneImage	Supported	Not supported
CiscoIPPhoneImageFile	Supported	Supported
CiscoIPPhoneGraphicMenu	Supported	Not supported
CiscoIPPhoneGraphicFileMenu	Not supported	Not supported
CiscoIPPhoneIconMenu	Supported	Not Supported
CiscoIPPhoneIconFileMenu	Supported	Not supported
CiscoIPPhoneStatus	Not supported	Not supported
CiscoIPPhoneStatusFile	Not supported	Not supported
CiscoIPPhoneExecute	Supported	Supported
CiscoIPPhoneResponse	Supported	Supported
CiscoIPPhoneError	Supported	Supported



**Note** Cisco Wireless Phone 840 or 860 support these objects only when they are in a file downloaded by retrieving a URI that is supplied as a CiscoIPPhoneExecuteItem in a CiscoIPPhoneExecute object.

## Multiplatform Phone Support for XML Applications and Services

The Cisco Small and Medium Business portfolio support Multiplatform phones. These phones are connected to a third-party call control system and support a subset of the CiscoIPPhone XML Objects. The following table shows the supported XML objects.

**Table 5: XML Objects Supported by Multiplatform Phones**

XML object	Multiplatform6800 Series	Multiplatform 7800 Series	Multiplatform 7832	Multiplatform 8800 Series (Except 8875)
CiscoIPPhoneMenu	Supported	Supported	Supported	Supported
CiscoIPPhoneText	Supported	Supported	Supported	Supported
CiscoIPPhoneInput	Supported	Supported	Supported	Supported
CiscoIPPhoneDirectory	Supported	Supported	Supported	Supported
CiscoIPPhoneImage	Supported	Supported	Supported	Supported

XML object	Multiplatform6800 Series	Multiplatform 7800 Series	Multiplatform 7832	Multiplatform 8800 Series (Except 8875)
CiscoIPPhoneImageFile	Not supported	Not supported	Not supported	Supported
CiscoIPPhoneGraphicMenu	Not supported	Not supported	Not supported	Not supported
CiscoIPPhoneGraphicFileMenu	Not supported	Not supported	Not supported	Not supported
CiscoIPPhoneIconMenu	Supported	Supported	Supported	Supported
CiscoIPPhoneIconFileMenu	Supported	Supported	Supported	Supported
CiscoIPPhoneStatus	Supported (Except 6841, 6821)	Supported (Except 7811, 7832)	Supported	Supported (Except 8832)
CiscoIPPhoneStatusFile	Supported (Except 6841, 6821)	Supported (Except 7811, 7832)	Supported	Supported (Except 8832)
CiscoIPPhoneExecute	Supported	Supported	Supported	Supported
CiscoIPPhoneResponse	Not supported	Not supported	Not supported	Not supported
CiscoIPPhoneError	Not supported	Not supported	Not supported	Not supported

## XML Object Definitions

The following sections provide definitions and descriptions of each CiscoIPPhone XML object.

### CiscoIPPhoneMenu

A menu on the phone contains a list of text items, one per line. Users choose individual menu items using the same mechanisms that are used for built-in menus in the phone.

When a menu loads, the phone behaves the same as for built-in phone menus. The user navigates through the list of menu items and eventually chooses one using either the Select softkey or the DTMF keys.

After the user chooses a menu option, the phone generates an HTTP request for the page with the URL or executes the uniform resource identifiers (URIs) that are associated with the menu item.

#### Related Topics

[Custom Client Services Overview](#)

### CiscoIPPhoneMenu 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>
```



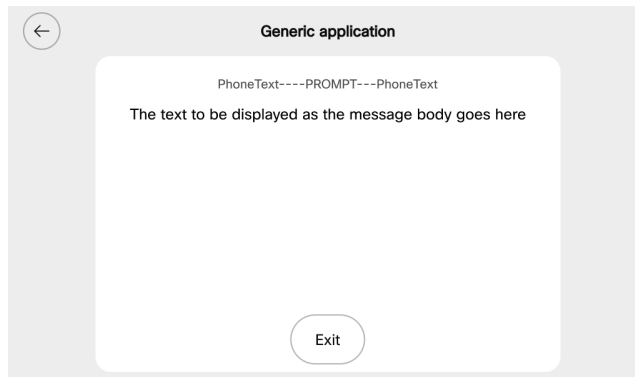
**Note** The *Name* field under the *<MenuItem>* supports a maximum of 64 characters. This field can also accept two carriage returns to allow the *MenuItem* name to span three lines on the display.

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. IP phones allow a maximum of 100 *MenuItems*. Each *<MenuItem>* includes a *Name* and an associated *URL*.

## CiscoIPPhoneText

The *CiscoIPPhoneText* XML object displays ordinary 8-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 IP phone firmware controls all other pagination and word wrap issues.

**Figure 1: Example of text message display on Cisco Video Phone 8875**



## CiscoIPPhoneText Definition

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

Two optional fields can appear in the XML message:

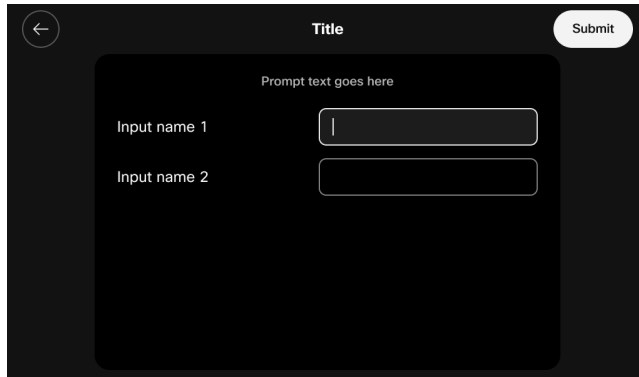
- The first optional field, *Title*, defines text that displays at the top of the display page. If a *Title* is not specified, the *Name* field of the last chosen *MenuItem* displays in the *Title* field.
- The second optional field, *Prompt*, defines text that displays at the bottom of the display page. If a *Prompt* is not specified, Cisco Unified Communications Manager clears the prompt area of the display pane.

Many XML objects that are described in this document also have *Title* and *Prompt* fields. These fields normally behave identically to behavior described in this section.

## CiscoIPPhoneInput

When an IP phone receives an XML object of type *CiscoIPPhoneInput*, it constructs an input form and displays it. The user enters data into each input item and sends the parameters to the target URL. The following figure shows a sample display that is receiving input from a user.

**Figure 2: Example of CiscoIPPhoneInput Object Display on Cisco Video Phone 8875**



Many XML objects that are described in this document also have *Title* and *Prompt* fields. These fields normally behave identically to behavior described in this section.



**Note Non-XML Text:** This document only describes the supported CiscoIPPhone XML objects. You can also deliver plain text using HTTP. Pages that are delivered as MIME type text/html behave exactly the same as XML pages of type *CiscoIPPhoneText*. One important difference is that you cannot include a title or prompt.



**Note Keypad navigation:** IP phones allow navigation to a specific line in a menu by pressing numeric DTMF keys. When a menu is on the display, the number for selecting the menu is on the left.

When normal text displays, the numbers do not display on the left side of the screen, but the navigation capability still exists. A carefully written text service display can take advantage of this capability.

During text entry, the phones display softkeys to assist users with text entry. Users can navigate between fields with the vertical scroll button that is used to navigate menus.

## CiscoIPPhoneInput Definition

```
<CiscoIPPhoneInput>
<Title>Directory title goes here</Title>
<Prompt>Prompt text goes here</Prompt>
<URL method="post">The target URL for the completed input goes here</URL>
<InputItem>
<DisplayName>Name of the input field to display</DisplayName>
<QueryStringParam>The parameter to be added to the target URL</QueryStringParam>
<DefaultValue>The default display name</DefaultValue>
<InputFlags>The flag specifying the type of allowable input</InputFlags>
</InputItem>
</CiscoIPPhoneInput>
```



The *Title* and *Prompt* tags in the object define text that are used in the same way as the identical fields in the other CiscoIPPhone XML objects.

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



**Note** The Cisco IP Phone 7800 and 8800 Series, Cisco IP Conference Phone 7832, and Cisco IP Conference Phone 8832 are the only phones that support the HTTP POST method.

The *InputItem* tag defines each item in the list. The number of *InputItems* must not exceed five. Each input item includes a *DisplayName*, which is the prompt that is written to the display for that particular item. Each item also has a *QueryStringParam*, which is the name of the parameter that is 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 attribute for each input item comprises a set of *InputFlags*. The following table describes the input types that are currently defined.

**Table 6: InputFlag Definitions**

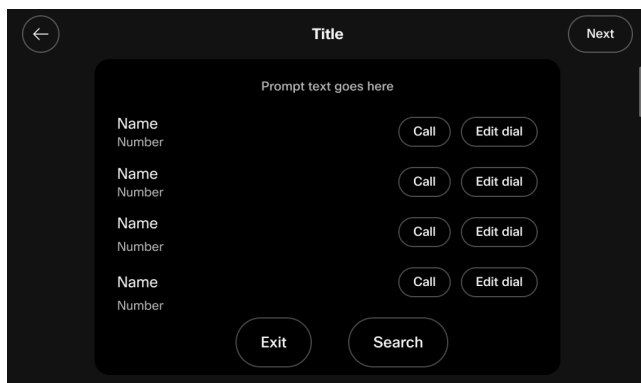
InputFlag	Description	Notes
A	Plain ASCII text	Use the DTMF keypad to enter text that consists of uppercase and lowercase letters, numbers, and special characters.
T	Telephone number	Enter only DTMF digits for this field. The acceptable input includes numbers, #, and *.
N	Numeric	Enter numbers as the only acceptable input.
E	Equation	Enter numbers and special math symbols. <b>Note</b> Not supported on the phones with Multiplatform Firmware.
U	Uppercase	Enter uppercase letters as the only acceptable input.
L	Lowercase	Enter lowercase letters as the only acceptable input.

InputFlag	Description	Notes
P	Password field	<p>Enter individual characters using the standard keypad-repeat entry mode. The system automatically converts accepted characters into an asterisk, keeping the entered value private.</p> <p><b>Note</b> P specifies the only <i>InputFlag</i> that works as a modifier. For example, specify a value of “AP” in the <i>InputFlag</i> field to use plain ASCII as the input type and to mask the input as a password by using an asterisk (*).</p>

## CiscoIPPhoneDirectory

The *CiscoIPPhoneDirectory* XML object supports the Directory operation of IP phones. The following figure shows how an XML *CiscoIPPhoneDirectory* object displays on the phone.

**Figure 3: Example of CiscoIPPhoneDirectory Object Display on Cisco Video Phone 8875**



## CiscoIPPhoneDirectory 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 IP phone displays the appropriate softkeys that are needed to dial the numbers that are listed on the display. The softkeys include the Edit Dial softkey, 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 Cisco Unified Communications Manager enterprise URL Directories parameter and CiscoIPPhone XML objects to display custom directories. The URL Directories parameter points to a URL that returns a *CiscoIPPhoneMenu* object to extend the directories menu. The request for URL Directories must return a valid *CiscoIPPhoneMenu* object, even if the object has no *DirectoryEntry* objects.

To create a custom directory, use the following optional objects in the order in which they are listed:

1. Use the *CiscoIPPhoneInput* XML object to collect search criteria.
2. Use the *CiscoIPPhoneText* XML object to display status messages or errors.
3. 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

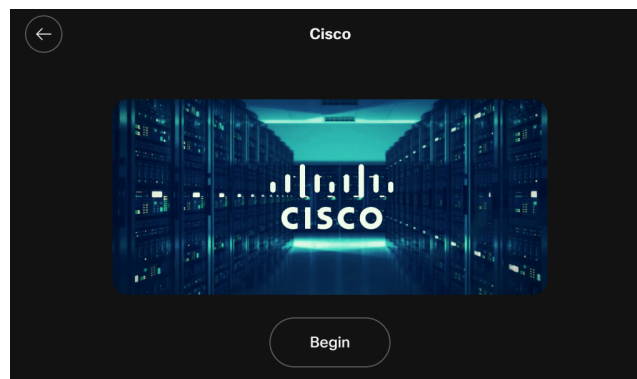
The *CiscoIPPhoneImage* provides a bitmap display with a 133 x 65 pixel pane (irrespective of the window mode being normal width or wide width), that is available to access services. Each pixel includes four grayscale settings. A value of three (3) displays as black, and a value of zero (0) displays as white.



**Note** The phone uses an LCD display, which inverts the palette.

The *CiscoIPPhoneImage* XML type lets you use the IP phone display to present graphics to the user.

**Figure 4: Example of CiscoIPPhoneImage Object Display on Cisco Video Phone 8875**



## CiscoIPPhoneImage Definition

```
<CiscoIPPhoneImage>
<Title>Image title goes here</Title>
```

```

<Prompt>Prompt text goes here</Prompt>
<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>
<SoftKeyItem>
<Name>Name of the soft key</Name>
<URL>URL of soft key</URL>
<Position>Numerical position of the soft key</Position>
</SoftKeyItem>
</CiscoIPPhoneImage>

```

The *Title* and *Prompt* elements serve the same purpose as they do in the other CiscoIPPhone 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 display. Position the upper, left corner of the graphic 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 display.

When you use CiscoIPPhoneImage with the Cisco Wireless IP Phone 8821, the phone ignores *LocationX* and *LocationY*. The image will always be placed in the center.

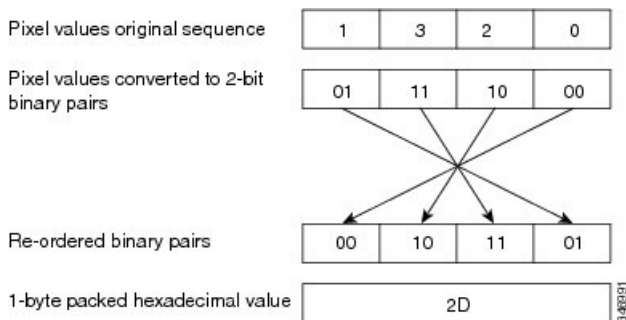
Use *Width* and *Height* to size the graphic. If the values do not match with the pixel stream specified in the *Data* field, results will be unpredictable or incorrect.

*Depth* specifies the number of bits per pixel. IP phones support a maximum value of 2 bits per pixel. A bit depth of 1 is black and white.

The *Data* tag delimits a string of hexadecimal digits that contain the packed value of the pixels in the display. In the IP phone, each pixel has only four possible values, which means that you can pack four pixels into a single byte. A pair of hexadecimal digits represents each byte.

The following figure 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 get converted to 2-bit binary numbers. Then, the binary pairs get reordered in sets of four to create a single reordered byte, which the two hexadecimal digits represent.

**Figure 5: Packed Pixel Translation Example**



## CiscoIPPhoneImage Example

The following XML code defines a *CiscoIPPhoneImage* object that displays the sequence of pixels shown in the above figure 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 comprises a contiguous stream of hexadecimal digits, with no spaces or other separators. If the number of pixels to be displayed does not represent an even multiple of four, pad the end of the pixel data with blank (zero value) pixels, so the data is packed correctly. The phone ignores the padded data.



**Note** Before displaying a graphic image on an IP phone, the software clears the pane dedicated to services. If a service has text or other information that must be preserved (including the title area), the information must get redrawn as part of the graphic. If the title is to be hidden, the graphic must be large enough to cover it.

## CiscoIPPhoneImageFile

To support these more advanced displays, the XML object allows the use of color PNG images in addition to the grayscale *CiscoIPPhoneImage* objects. The *CiscoIPPhoneImageFile* object behaves like the *CiscoIPPhoneImage* object, except for the image data. Instead of using the `<Data>` tag to embed the image data, the `<URL>` tag points to the PNG image file.

The web server must deliver the PNG image to the phone with an appropriate MIME Content-Type header, such as `image/png`, so that the phone recognizes the content as a compressed, binary PNG image. The PNG image can be either palettized or RGB, and the maximum image size and color depth are model dependent (see the following table).

In the following table, the specifications are the same for On-Premise and Multiplatform phone firmware.

Model	Resolution (see note 1) (width x height)	Resolution in wide mode (width x height)	Color, Grayscale, Monochrome	Color depth (bits)
Cisco IP Phone 6800 Series	N/A	N/A	Grayscale	—
Cisco IP Phones 7811, 7821, 7841, 7861 (see Note 3)	N/A	N/A	Monochrome	—
Cisco IP Conference Phone 7832 (see Note 3)	N/A	N/A	Grayscale	—
Cisco IP Phone 8811	559 x 265	N/A	Monochrome	0-10

Model	Resolution (see note 1) (width x height)	Resolution in wide mode (width x height)	Color, Grayscale, Monochrome	Color depth (bits)
Cisco Unified IP Conference Station 8831	396 x 162	N/A	Monochrome	—
Cisco IP Conference Phone 8832	480x128	N/A	Color	24
Cisco IP Phone 8841, 8845, 8851, 8851NR, 8861, 8865, and 8865NR	559 x 265	N/A	Color	24
Cisco Video Phone 8875, 8875NR	696x312	696x 456	Color	24
Cisco Wireless IP Phone 8821	240 x 215	N/A	Color	24
Cisco Desk Phone 9841 (On-Premise Only)	320 x 108	320 x 144	Grayscale	—
Cisco Desk Phone 9851 (On-Premise Only)	600 x 280	600 x 328	Color	24



- Note**
- Resolution represents the size of the display that is accessible by Services; not the full resolution of the physical display.
  - The Cisco IP Phone 7800 Series and Cisco IP Conference Phone 7832 do not support CiscoIPPhoneImageFile.

If the number of colors in the image is not reduced to match the phone capabilities, the image will be dithered by the phone and yield less than desirable results in most cases. To reduce the number of colors in a graphics editing program, such as Adobe Photoshop, use the *Posterize* command. The *Posterize* command takes one value as input for the number of color tones per color channel.

The following figure shows a CiscoIPPhoneImageFile object on phone display.

**Figure 6: Example of CiscoIPPhoneImageFile Object Display on Cisco Video Phone 8875**



**Note** For Cisco Wireless phone 840/860, <LocationX> and <LocationY> attributes are ignored. The image is centered on the display.

## CiscoIPPhoneImageFile Definition

```
<CiscoIPPhoneImageFile>
  <Title>Image Title goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <LocationX>Horizontal position of graphic</LocationX>
  <LocationY>Vertical position of graphic</LocationY>
  <URL>Points to the PNG image</URL>
</CiscoIPPhoneImageFile>
```

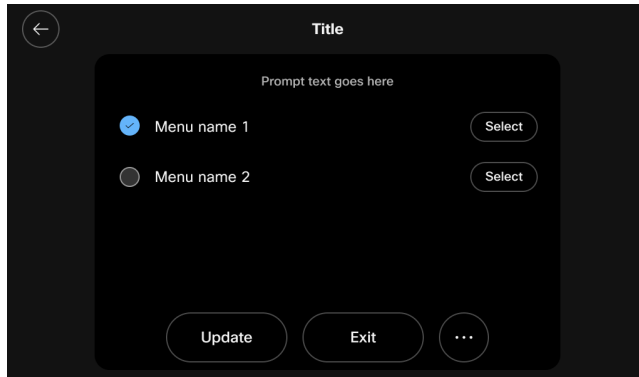
## CiscoIPPhoneIconMenu

Icon menus serve the same purpose as text menus: they allow a user to select a URL from a list. Use icon menus in situations when you want to provide additional visual information to the user to show the state or category of an item. For example, you include a read and unread icon in a mail viewer. You can use the icons can to convey the message state.

Icons in the *CiscoIPPhoneIconMenu* object have a maximum width of 16 pixels and a maximum height of 10 pixels.

The following figure shows an IconMenu on an IP phone.

Figure 7: Example of CiscoIPPhoneIconMenu on Cisco Video Phone 8875



The system presents the information as a bitmap graphic to the left of the menu item text. The user selects menu items in the same way as a *CiscoIPPhoneMenu* object.

### Related Topics

[CiscoIPPhoneMenu](#), on page 6

[CiscoIPPhoneImage](#), on page 11

## CiscoIPPhoneIconMenu Definition

```
<CiscoIPPhoneIconMenu>
  <Title>Title text goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <MenuItem>
    <IconIndex>Indicates what IconItem to display</IconIndex>
    <Name>The name of each menu item</Name>
    <URL>The URL associated with the menu item</URL>
  </MenuItem>
  <SoftKeyItem>
    <Name>Name of softkey</Name>
    <URL>URL or URI of softkey</URL>
    <Position>Position information of the softkey</Position>
  </SoftKeyItem>
  <IconItem>
    <Index>A unique index from 0 to 9</Index>
    <Height>Size information for the icon</Height>
    <Width>Size information for the icon</Width>
    <Depth>Number of bits per pixel</Depth>
    <Data>Packed Pixel Data</Data>
  </IconItem>
</CiscoIPPhoneIconMenu>
```

The XML tags in *CiscoIPPhoneIconMenu* use the tag definitions for *CiscoIPPhoneImage* and *CiscoIPPhoneMenu*. Although the semantics of the tags are identical, you can have only 32 *MenuItem* objects in a *CiscoIPPhoneIconMenu* object.

### Related Topics

[CiscoIPPhoneMenu](#), on page 6

[CiscoIPPhoneImage](#), on page 11



## CiscoIPPhoneIconFileMenu

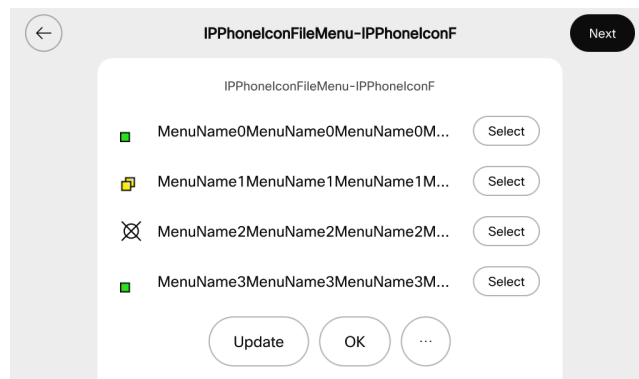


**Note** The *CiscoIPPhoneIconFileMenu* object is updated to support new attributes. For details, see [Enhanced Icon Menu Support Feature, on page 18](#).

This icon menu is similar to *CiscoIPPhoneMenu*, but it uses color PNG icons rather than grayscale CIP icons. Use icon menus in situations when you want to provide additional visual information to the user to show the state or category of an item. For example, you can use icons to indicate priority (see the following figure).

Icons in the *CiscoIPPhoneIconFileMenu* object have a maximum width of 18 pixels and a maximum height of 18 pixels. Instead of using the `<Data>` tag to embed the image data into the `<IconItem>` tag, this object uses a `<URL>` tag to point to the PNG image file to be used for that icon.

**Figure 8: Example of CiscoIPPhoneIconFileMenu Object Display on Cisco Video Phone 8875**



## CiscoIPPhoneIconFileMenu Definition

```
<CiscoIPPhoneIconFileMenu>
  <Title>Title text goes here</Title>
  <Prompt>Prompt text goes here</Prompt>
  <MenuItem>
    <IconIndex>Indicates what IconItem to display</IconIndex>
    <Name>The name of each menu item</Name>
    <URL>The URL associated with the menu item</URL>
  </MenuItem>
  <IconItem>
    <Index>A unique index from 0 to 9</Index>
    <URL>location of the PNG icon image</URL>
  </IconItem>
</CiscoIPPhoneIconFileMenu>
```



**Note** The Cisco Unified IP Phone 6900 Series do not display the *Title* and *Prompt* menu fields at the same time. If both *Title* and *Prompt* fields are defined at the same time, then these phones display only the *Prompt* field.

## Enhanced Icon Menu Support Feature

The Enhanced Icon Menu Support feature extends the existing *CiscoIPPhoneIconFileMenu* XML object by allowing:

- An icon in its *<Title>* element.
- Internal phone firmware icons, like security state or call state icons, in its *<MenuItems>* and *<Title>* elements.

### Supported IP Phones and Codecs

The following table lists the IP phone models that support the Enhanced Icon Menu Support feature.

Phone model	Support	Firmware supported
Cisco Unified IP Phone 9900 Series		
9971	Supported	9.0(1) and later
9951	Supported	9.0(1) and later
Cisco IP Phone 8800 Series (On-premise and Multiplatform Phones)		
8811	Not supported	—
8841	Not supported	—
8845	Not supported	—
8851	Not supported	—
8851NR	Not supported	—
8861	Not supported	—
8865	Not supported	—
8865NR	Not supported	—
Cisco Video Phone 8875 (On-premise and Multiplatform Phones)		
8875	Not supported	—
8875NR	Not supported	—
Cisco IP Conference Phones 8830 Series (On-premise and Multiplatform Phones)		
8831	Not supported	—
8832	Not supported	—
Cisco Wireless IP Phone 8820 Series		
8821	Not supported	—
Cisco IP Phone 7800 Series (On-premise and Multiplatform Phones)		

Phone model	Support	Firmware supported
7811	Not supported	—
7821	Not supported	—
7841	Not supported	—
7861	Not supported	—
Cisco IP Conference Phone 7830 Series (On-premise and Multiplatform Phones)		
7832	Not supported	—
Cisco IP Phone 6800 Series (Multiplatform Phones)		
6841	Not supported	—
6851	Not supported	—



**Note** Cisco recommends the use of latest firmware. The firmware can be downloaded from the following location (requires login or service contract):

<http://software.cisco.com/download/navigator.html?i=!mmd>

### CiscoIPPhoneIconFileMenu XML Object Changes

The following changes have been made in the CiscoIPPhoneIconFileMenu XML object for the Enhanced Icon Menu Support feature:

- The CiscoIPPhoneIconFileMenu schema is updated to allow an *IconIndex* attribute in the `<Title>` element.
- A Resource URI attribute is available for the `<URL>` element of the `<IconItem>` element. This Resource URI can be used in place of the HTTP URL.
- The Resource URI identifies the icons in the `<IconItems>`. When a phone parses the `<URL>` element in `<IconItem>`, the phone looks for the Resource URI.
  - If the Resource URI is present, the phone validates the URI against the valid Resource Icon values. If the validation is successful, the phone uses the icon specified by the Resource URI.
  - If the Resource URI is not present or if the URI fails the validation against a recognized Icon value, then a default unknown-icon image displays.

### Related Topics

[Valid Resource Icon Names](#), on page 20

### Schema Definition

The definition of the CiscoIPPhoneIconFileMenu schema remains the same except for the `<Title>` element and the *IconIndex* attribute specified as follows:

```
<xsd:complexType name="Title">
  <xsd:attribute name="IconIndex"
```

```

    type="xsd:unsignedShort"
    use="optional"/>
</xsd:complexType>

```

### CiscoIPPhoneIconFileMenu Example

The following is an example of the CiscoIPPhoneIconFileMenu object with IconIndex attribute in <Title> element and Resource URI attribute in <IconItem> element:

```

<CiscoIPPhoneIconFileMenu>
  <Title IconIndex="2">Conference List</Title>
  <IconItem>
    <Index>1</Index>
    <URL>Resource:Icon.SecureCall</URL>
  </IconItem>
  <IconItem>
    <Index>2</Index>
    <URL>Resource:Icon.Connected</URL>
  </IconItem>
  <IconItem>
    <Index>3</Index>
    <URL>Resource:AnimatedIcon.Ringin</URL>
  </IconItem>
  <MenuItem>
    <Name>Schmo, Joe</Name>
    <IconIndex>1</IconIndex>
    <URL>http://192.168.1.12:8080/details?user=jschmo</URL>
  </MenuItem>
  <MenuItem>
    <Name>Blow, Joe</Name>
    <IconIndex>2</IconIndex>
    <URL>http://192.168.1.12:8080/details?user=jblow</URL>
  </MenuItem>
  <MenuItem>
    <Name>Joining, Just Now</Name>
    <IconIndex>3</IconIndex>
    <URL>http://192.168.1.12:8080/details?user=jjoining</URL>
  </MenuItem>
</CiscoIPPhoneIconFileMenu>

```

### Valid Resource Icon Names

The following are the valid Resource Icon names:

- Icon.Connected
- Icon.AuthenticatedCall
- Icon.SecureCall
- Icon.OnHook
- Icon.OffHook
- Icon.Messages
- Icon.InUse
- Icon.Headset
- Icon.Handset
- Icon.Speaker

- Icon.Locked
- Icon.UnLocked
- Icon.Checked
- Icon.UnChecked
- Icon.RadioButtonOn
- Icon.RadioButtonOff
- AnimatedIcon.Ringin
- AnimatedIcon.Hold
- AnimatedIcon.MessageWaiting
- AnimatedIcon.StreamingRx
- AnimatedIcon.StreamingTx
- AnimatedIcon.StreamRxTx
- AnimatedIcon.Throbber

### Troubleshooting CiscoIPPhoneIconFileMenu XML Objects Using Enhanced Icon Menu Support Feature

The following errors and conditions may occur in the Enhanced Icon Menu Support feature:

- If the CiscoIPPhoneIconFileMenu object is invalid, a parsing error is generated and a CiscoIPPhoneError object (with Number="1") is returned as the response.
- If the Resource URI does not specify a recognized Icon resource, then a default unknown-icon image is displayed.

#### Error Handling

Standard XML services debugging techniques are applied to the Enhanced Icon Menu Support feature. The root cause for any parsing errors displays in the phone console logs. For HTTP requests and responses, sniffer traces and web server debug can be used to examine the CiscoIPPhoneIconFileMenu object to ensure that the object conforms to the schema.

## CiscoIPPhoneStatus

The *CiscoIPPhoneStatus* object is also a displayable object, but differs from other objects in that it displays on the Call plane of the phone rather than the Services plane. The CiscoIPPhoneStatus object hovers above the Call plane and is typically used in conjunction with CTI applications to present application status to the user.

The Status object cannot be closed or cleared by the user (for example, by pressing Services) because the Status object is only present on the Call plane. In order to clear the object, the phone must execute the Init:AppStatus URI. This would typically occur as the result of an application server pushing an Execute object to the phone that contains the Init:AppStatus URI.




---

**Note** The CiscoIPPhoneStatus object can only be pushed (HTTP POST) to the phone; the object cannot be pulled (HTTP GET).

---

The *CiscoIPPhoneStatus* object can be refreshed or replaced at any time. It is not necessary to clear an existing Status object before sending a new Status object. The new object simply replaces the old object.

The following figure shows the CiscoIPPhoneStatus object that contains the following visual elements:

- 106 x 21 graphics area for displaying CIP images (same image format as CiscoIPPhoneImage)
- Seedable, free-running timer (optional)
- Single-line text area (optional)




---

**Note** The Cisco Video Phone 8875, Cisco Desk Phone 9800 Series, and the Cisco IP Phone 8821 do not support CiscoIPPhoneStatus.

---

## CiscoIPPhoneStatus Definition

```
<CiscoIPPhoneStatus>
  <Text>This is the text area</Text>
  <Timer>Timer seed value in seconds</Timer>
  <LocationX>Horizontal alignment</LocationX>
  <LocationY>Vertical alignment</LocationY>
  <Width>Pixel width of graphic</Width>
  <Height>Pixel height of graphic</Height>
  <Depth>Color depth in bits</Depth>
  <Data>Hex binary image data</Data>
</CiscoIPPhoneStatus>
```

## Dynamic Application Status Window Size

You can enable applications to dynamically adjust their window sizes based on the displayed content. The minimum size requirements limit the windows size so that it is large enough to stand out from the Overview content. For example, using a smaller window for an application allows more content from the Overview to be displayed. Sizing the window occurs when the phone receives a CiscoIPPhoneStatus or CiscoIPPhoneStatusFile object with its associated PNG file.

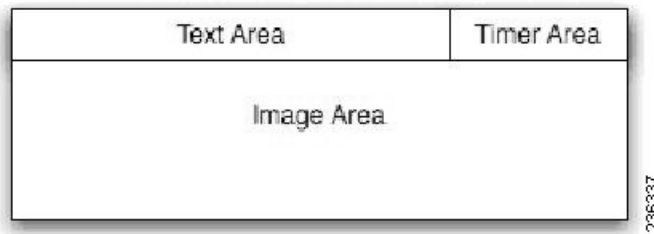
The following phones do not support the Application Status window:

- Cisco IP Phone 6800 Series (Multiplatform Phones)
- Cisco IP Phone 7800 Series (On-premise and Multiplatform Phones)
- Cisco IP Conference Phone 7832 (On-premise and Multiplatform Phones)
- Cisco Wireless IP Phone 8821 and 8821-EX
- Cisco Unified IP Conference Phone 8831 (On-premise and Multiplatform Phones)
- Cisco Video Phone 8875 (On-premise and Multiplatform Phones)
- Cisco Desk Phone 9800 Series (On-premise and Multiplatform Phones)

The Application Status window contains three main areas (see the following figure):

- Text Area
- Timer Area
- Image Area

**Figure 9: Application Status Window Elements**



**Note** Self terminating XML elements, undeclared or missing elements, and elements with the default values are all considered unconfigured elements.

To allow dynamic sizing, do not configure the Text and Timer areas with any value other than the default used by the XML parser. If both elements are not configured, you can proceed, but must follow these rules:

- Do not display the Text Area and Timer Area sections of the Application Status window.
- If the LocationX element is not configured or is set to centered, and the image provided is less than the maximum width allowed, the Image Area can be resized.
- If the image provided is smaller than the minimum width, the minimum allowed window width should be used.
- If the width of the image provided is between the minimum and maximum sizes of the window, the window should be sized to display the image as well as the standard surrounding borders.
- The image height should never change.

See the following table for an overview of the maximum and minimum image area sizes by phone model. Most phone models support all sizes between the minimum and maximum.

**Table 7: Application Status Window Allowable Image Sizes**

Phone models	Maximum image area width	Minimum image area width	Maximum image area height
8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR	414	70	70

The following table shows an overview of the text and timer area sizes by phone model.

Table 8: Application Status Window Allowable Text and Timer Sizes

Phone models	Text area size (WxH)	Timer area size (WxH)	Text area size No timer (WxH)
8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR	300x36	100x36	414x36

## CiscoIPPhoneStatusFile

The behavior of this object is identical to the *CiscoIPPhoneStatus* object, except it uses a color PNG image instead of a grayscale CIP image for the graphics area.

The Cisco Video Phone 8875, Cisco Desk Phone 9800 Series, and the Cisco IP Phone 8821 do not support CiscoIPPhoneStatusFile.

### Related Topics

[CiscoIPPhoneStatus](#), on page 21

## CiscoIPPhoneStatusFile Definition

```
<CiscoIPPhoneStatusFile>
<Text>This is the text area</Text>
<Timer>Timer seed value in seconds</Timer>
<LocationX>Horizontal alignment</LocationX>
<LocationY>Vertical alignment</LocationY>
<URL>location of the PNG image</URL>
</CiscoIPPhoneStatusFile>
```

Note that instead of using the *<Data>* tag to embed the image data, this object uses a *<URL>* tag to point to the PNG image file to be used for the graphics area.

### Related Topics

[Dynamic Application Status Window Size](#), on page 22

## CiscoIPPhoneExecute

The *CiscoIPPhoneExecute* object differs from the other CiscoIPPhone objects. It is not a displayable object for providing user interaction. The purpose of this object is to deliver (potentially multiple) execution requests to the phone.

Like the other XML objects, the CiscoIPPhoneExecute can be either pushed (HTTP POST) or pulled (HTTP GET). Upon receiving a CiscoIPPhoneExecute object, the phone begins executing the specified ExecuteItems. Order of execution is not guaranteed, so ExecuteItems will likely not execute in the order in which they are listed in the CiscoIPPhoneExecute object.




---

**Note** Limit the requests to three ExecuteItems: only one can be a URL and two URIs per *CiscoIPPhoneExecute* object, or you can send three URIs with no URL.

---





**Note** For Cisco Wireless Phone 840/860

If the phone is locked, the notification is displayed. If the phone is unlocked or has not been unlocked after a reboot, the phone display a notification that does not interrupt the user when they receive a request. The user can choose when to run the request. In case of Unicast/Multicast RTP requests, the request runs if the phone is idle and otherwise the user will have the opportunity to play the request or ignore it.

## CiscoIPPhoneExecute Definition

```
<CiscoIPPhoneExecute>
  <ExecuteItem URL="the URL or URI to be executed"/>
</CiscoIPPhoneExecute>
```

The *<ExecuteItem>* tag of the *CiscoIPPhoneExecute* object includes an optional attribute called *Priority*. The *Priority* attribute is used to inform the phone of the urgency of the execute request and to indicate whether the phone should be interrupted to perform the request. The *Priority* levels determine whether the phone must be idle to perform the requested action. The *Idle Timer* (along with an optional *Idle URL*) is defined globally in the Cisco Unified Communications Manager Administration Enterprise Parameters and can be overridden on an individual phone basis in the Cisco Unified Communications Manager Device configuration.

The following table lists the *Priority* levels and their behavior.

Priority	Behavior	Description
0	Execute Immediately	The URL executes regardless of the state of the phone. If the <i>Priority</i> attribute does not get specified in the <i>&lt;ExecuteItem&gt;</i> , the default priority gets set to zero for backward compatibility.
1	Execute When Idle	The URL gets delayed until the phone goes idle, then it executes.
2	Execute If Idle	The URL executes on an idle phone; otherwise, it does not get executed (it does not get delayed).



**Note** The *Priority* attribute is only used for HTTP URLs. Internal URIs always execute immediately.

## CiscoIPPhoneExecute Example

The following *CiscoIPPhoneExecute* object results in the phone playing an alert chime, regardless of the state of the phone, but waits until the phone goes idle before displaying the specified XML page.

```
<CiscoIPPhoneExecute>
  <ExecuteItem Priority="0" URL="Play:chime.raw"/>
  <ExecuteItem Priority="1" URL="http://server/textmessage.xml"/>
</CiscoIPPhoneExecute>
```

## Create a Remote Problem Report with CiscoIPPhoneExecute

You can use the *CiscoIPPhoneExecute* object to generate a problem report. This report has the same content as the report generated by the Problem Report Tool (PRT).



**Note** Only the Cisco IP Wireless Phone 8821 supports this function.

## Procedure

**Step 1** Push the following XML to the phone to generate the problem report.

```
<CiscoIPPhoneExecute>
<ExecuteItem Priority="0" URL="Device:GeneratePRT"/>
</CiscoIPPhoneExecute>
```

The phone returns one of these responses:

- ```
<CiscoIPPhoneResponse>
<ResponseItem URL="Device:GeneratePRT" Data="Success" Status="0"/>
</CiscoIPPhoneResponse>
```

This means that the request was successful, and the Device:GeneratePRT command has been accepted.

- ```
<CiscoIPPhoneResponse>
<ResponseItem URL="Device:GeneratePRT" Data="There is pending PRT" Status="6"/>
</CiscoIPPhoneResponse>
```

This means that the request failed because there is pending problem report request, likely requested from the phone.

**Step 2** If the phone returns a Success message, then poll the problem report creation with this XML:

```
<CiscoIPPhoneExecute>
<ExecuteItem Priority="0" URL="Device:PRTStatus"/>
</CiscoIPPhoneExecute>
```

The phone returns one of these responses:

- ```
<CiscoIPPhoneResponse>
<ResponseItem URL="Device:PRTStatus" Data="Generating PRT" Status="0"/>
</CiscoIPPhoneResponse>
```

This means that the problem report creation is in progress.

- ```
<CiscoIPPhoneResponse>
<ResponseItem URL="Device:PRTStatus" Data="There is no pending PRT invoked from XSI"
Status="6"/>
</CiscoIPPhoneResponse>
```

This means that the phone didn't receive the problem report request.

**Step 3** Continue polling until you get the message:

```
<CiscoIPPhoneResponse>
<ResponseItem URL="Device:PRTStatus" Data="Generated PRT at
https://xx.xx.xx.xx/FS/prt-yyyyymmdd-hhmmss-xxxxxxxxxxxxx.tar.gz"
Status="0"/>
</CiscoIPPhoneResponse>
```

Where:

- xx.xx.xx.xx is the IP address of the phone.

- prt-yyyyymmdd-hhmmss-xxxxxxxxxxxx of the date (yyyymmdd), time (hhmmss), and MAC address (xxxxxxxxxxxx) of the phone.

**Step 4** Access the URL and download the problem report.

## CiscoIPPhoneResponse

The *CiscoIPPhoneResponse* objects provide messages and information resulting from a *CiscoIPPhoneExecute*. As a result, a *ResponseItem* exists for each *ExecuteItems* that you send. The order differs based on completion time, and the execution order is not guaranteed.



**Note** The Cisco IP Phone 6800 Series, Cisco IP Phone 7800 Series, and Cisco IP Phone 8800 Series Multiplatform Phones do not support CiscoIPPhoneResponse.

The URL attribute specifies the URL or URI that was sent with the request. The Data attribute contains any special data for the item. The Status attribute specifies a status code. Zero indicates that no error occurred during processing of the *ExecuteItem*. If an error occurred, the phone returns a *CiscoIPPhoneError* object.

### CiscoIPPhoneResponse Definition

```
<CiscoIPPhoneResponse>
  <ResponseItem Status="the success or failure of the action"
    Data="the information returned with the response"
    URL="the URL or URI specified in the Execute object"/>
</CiscoIPPhoneResponse>
```

## CiscoIPPhoneError

The following list gives possible *CiscoIPPhoneError* codes:

- Error 1 = Error parsing *CiscoIPPhoneExecute* object
- Error 2 = Error framing *CiscoIPPhoneResponse* object
- Error 3 = Internal file error
- Error 4 = Authentication error

### CiscoIPPhoneError Definition

```
<CiscoIPPhoneError Number="x"/> optional error message <CiscoIPPhoneError>
```

The text value of the *CiscoIPPhoneError* object may contain an optional error message to further describe the nature of the error condition.

## Custom Softkeys

IP Phones can use custom softkeys with any of the displayable CiscoIPPhone XML objects, with the following exceptions:

- *CiscoIPPhoneStatus* object, which cannot control softkeys
- *CiscoIPPhoneExecute* object, which is not displayable

Softkeys can have either URL or URI actions associated with them. The *SoftkeyItem* can define separate actions to be taken when the softkey is pressed and released. The standard UI behavior is to execute an action when a key is released, and this action is defined by the `<URL>` tag. An action can also be taken when the softkey is initially pressed by including the optional `<URLDown>` tag. For example, you might use `<URLDown>` for a press-to-talk application in which pressing the button starts audio streaming and releasing the button stops it.




---

**Note** The `<URLDown>` tag can only contain Internal URIs: it cannot contain an HTTP URL. The “URL” in the name “URLDown” does not signify that an HTTP URL can be used.

---

The API does not support the use of *SoftKeyItems* to enclose multiple *SoftKeyItem* entries.

## SoftKeyItem Definition

```
<SoftKeyItem>
  <Name>Displayed sofkey label</Name>
  <URL>URL or URI action for softkey RELEASE event</URL>
  <URLDown>URL or URI action for softkey PRESS event</URLDown>
  <Position>position of softkey</Position>
</SoftKeyItem>
```

### Position

- -1 designates the Application/Settings button
- Cisco Wireless Phone 8821: 1 to 8 designates the softkeys
- Other phones: 1 to 16 designates the softkeys

The *SoftKeyItem* in the -1 position does not display on the phone screen. If the user pressed the Application or Settings button while in an XSI application, the action associated with the -1 position executes.

For the Cisco Wireless IP Phone 8821, the softkey position 1 corresponds to the right softkey. All the other positions correspond to the left softkey.

## SoftKeyItem Example 1

In this example, a *CiscoIPPhoneText* object has a single custom softkey defined.

```
<CiscoIPPhoneText>
  <Text>This object has one softkey named "Custom"</Text>
  <SoftKeyItem>
    <Name>Custom</Name>
```

```

    <URL>http://someserver/somepage</URL>
    <Position>4</Position>
  </SoftKeyItem>
</CiscoIPPhoneText>

```

If any custom softkeys are defined in the XML object, then all default softkeys are removed from that object. To retain default softkey behavior, you must explicitly define the softkeys in the XML object using a `<SoftKeyItem>` tag. The internal Softkey URIs can be used in the `<URL>` tag of `<SoftKeyItem>` to invoke default softkey actions from custom softkeys.



**Note** If there are no custom softkeys and there is no default softkey placed in position 1, either a Next or Update softkey is assigned automatically. If the URL is a Refresh URL, the Next softkey is assigned. If not, the Update softkey is assigned.

### Related Topics

[Internal URI Features](#)

## SoftKeyItem Example 2

The following softkey definitions would provide the Custom softkey, without losing the default Select softkey behavior.

```

<SoftKeyItem>
  <Name>Select</Name>
  <URL>SoftKey:Select</URL>
  <Position>1</Position>
</SoftKeyItem>
<SoftKeyItem>

```

## XML Considerations

The XML parser in the IP Phones does not function as a fully-capable XML parser. Do not include any tags other than those defined in your XML display definitions.



**Note** All CiscoIPPhone element names and attribute names are case sensitive.

## Mandatory Escape Sequences

By XML convention, the XML parser also requires that you provide escape values for a few special characters. The following table lists characters and their escape values.

**Table 9: Escape Sequences for Special Characters**

Character	Name	Escape sequence
&	Ampersand	&amp;
"	Quote	&quot;

Character	Name	Escape sequence
'	Apostrophe	&apos;
<	Left angle bracket	&lt;
>	Right angle bracket	&gt;

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

## XML Encoding

Because the phone firmware can support multiple encodings, the XML encoding should always be set in the XML header.

If the XML encoding header is not specified, the phone will default to the encoding specified by the current user locale.



**Note** This behavior is NOT compliant with XML standards, which specify UTF-8 as the default encoding, so any UTF-8 encoded XML object must have the encoding explicitly set for the phone to parse it correctly.

The encoding value specified in the XML header must match one of the encodings provided by the IP Phone in its Accept-Charset HTTP request header, as shown in [XML Encoding Example](#), on page 30

## XML Encoding Example

The following examples illustrate UTF-8 and ISO-8859-1 encoding, respectively:

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

### Related Topics

[HTTP Encoding Header Setting](#)

## Application Event Handlers

The Application Manager API includes an Application Management Event Handler, which is supported by any displayable object, as noted in the following table. The unsupported objects are not contained in a standard application context and are handled differently by the Application Manager API.



**Note** The Multiplatform phones do not support the Application Event Handlers.

*Table 10: Application Event Handler Support*

Supported	Unsupported
CiscoIPPhoneMenu	CiscoIPPhoneStatus

Supported	Unsupported
CiscoIPPhoneText	CiscoIPPhoneStatusFile
CiscoIPPhoneInput	
CiscoIPPhoneDirectory	
CiscoIPPhoneImage	
CiscoIPPhoneImageFile	
CiscoIPPhoneGraphicMenu	
CiscoIPPhoneGraphicFileMenu	
CiscoIPPhoneIconMenu	
CiscoIPPhoneIconFileMenu	



**Note** Support for the Application Event Handlers requires an updated XML Parser.

#### Related Topics

[Application](#)

[Supported Phone Models](#)

[Updated XML Parser and Schema Enforcement](#)

## Application Event Handler Attributes

The Application Event Handlers can be attached to a supported object by specifying the attributes described in the following table.



**Note** An Application URI with Priority=0 is not allowed in the Application Event Handlers.

*Table 11: Application Event Handler Attributes*

Attribute	Description
appID	Identifies the application to which this displayable object belongs. The format of the appID attribute should be in the format vendor/product, such as Cisco/Unity, but this syntax is not enforced, and the application can assign any unique identifier.

Attribute	Description
onAppFocusLost	<p>Invoked when the application loses focus, if one of the following conditions occurs:</p> <ul style="list-style-type: none"> <li>• The application context loses focus</li> <li>• The application was navigated away from, either directly by the user, or programmatically by a refresh header or HTTP push</li> </ul> <p><b>Note</b> If a Notify URI is used as the event handler, a notification is sent with this default data: <code>&lt;notifyApplicationEvent appId="appId" type="focusLost"/&gt;</code></p>
onAppFocusGained	<p>Invoked when the application gains focus, if one of the following conditions occurs:</p> <ul style="list-style-type: none"> <li>• The application is Active and the application context gains focus</li> <li>• The application was navigated to, either directly by the user, or by a refresh header or HTTP push</li> </ul> <p>If a Notify URI is used as the event handler, a notification is sent with this default data: <code>&lt;notifyApplicationEvent appId="appId" type="focusGained"/&gt;</code></p>
onAppMinimized	<p>Invoked when the application is minimized.</p> <p>An application can only be minimized in a program by a call to <code>App.Minimize</code>, but this invocation could occur by direct action of the user (for example, from a softkey invocation) or from the application using a push request. <code>&lt;notifyApplicationEvent appId="appId" type="minimized"/&gt;</code></p>
onAppClosed	<p>Invoked when the application closes, if one of the following conditions occur:</p> <ul style="list-style-type: none"> <li>• The application context is closed which will, in turn, close all applications in its stack</li> <li>• The application no longer exists on the context URL stack because it was navigated away from, or because it was pruned from the URL stack (stack size exceeded)</li> </ul> <p><b>Note</b> This event handler cannot contain HTTP or HTTPS URLs.</p> <p><b>Note</b> If a Notify URI is used as the event handler, a notification is sent with this default data: <code>&lt;notifyApplicationEvent appId="appId" type="closed"/&gt;</code></p>

**Related Topics**[Application](#)



## Event Handler Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="notifyApplicationEvent">
    <xs:complexType>
      <xs:attribute name="appId" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="64"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="closed"/>
            <xs:enumeration value="minimized"/>
            <xs:enumeration value="focusLost"/>
            <xs:enumeration value="focusGained"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

## Event Handler Example

```

<CiscoIPPhoneImage appId="Cisco/Unity"
  onAppFocusLost="RTPRx:Stop; RTPTx:Stop; Notify:http:server:80:path"
  onAppFocusGained="http://server/mainpage/updateUI"
  onAppClosed="Notify:http:server:80:eventlistener/appClosed">
  ...
</CiscoIPPhoneImage>

```

