



CHAPTER 3

Internal URI Features

Internal uniform resource identifiers (URIs) provide access to embedded phone features such as placing calls, playing audio files, and invoking built-in object features. These sections provide details about the available internal URIs:

- [Supported URIs by Phone Model](#)
- [URIs for Pressing Buttons on the Phone](#)
- [URIs for Invoking Softkey Functionality](#)
- [URIs to Control RTP Streaming](#)
- [Miscellaneous URIs](#)

Supported URIs by Phone Model

[Table 3-1](#) lists the URIs that are supported for Release 6.0(1).

Table 3-1 URIs Supported for Release Cisco Unified IP Phone Services SDK

URI	7905G, 7911G 7912G, 7931G	7920G	7921G	7940G 7960G	7941G/7941G-GE, 7961G/7961G-GE, 7942G, 7962G, 7945G, 7965G, IP Communicator	7970G 7971G-GE 7975G
Key	X	X	X	X	X	X
Softkey	X	X	X	X	X	X
Init	X	X	X	X	X	X
Dial, EditDial	X	X	X	X	X	X

Table 3-1 URIs Supported for Release Cisco Unified IP Phone Services SDK (continued)

URI	7905G, 7911G 7912G, 7931G	7920G	7921G	7940G 7960G	7941G/7941G-GE, 7961G/7961G-GE, 7942G, 7962G, 7945G, 7965G, IP Communicator	7970G 7971G-GE 7975G
Play	X	X	X	X	X	X
QueryStringParam	X	X	X	X	X	X
Unicast RTP	X	X ^{1,2}	X	X	X	X
Multicast RTP	X	X	X	X	X	X
Display	—	—	—	—	—	X
Vibrate	—	X	X	—	—	

1. Does not support the Volume parameter for RTP Receive streams.
2. Cisco Unified IP Phone 7920G only supports one incoming and one outgoing unicast stream.

URIs for Pressing Buttons on the Phone

The Key URI allows a programmer to send an event that a key has been pressed. The system initiates the event as if the button was physically pressed.

Note that when buttons are pressed with this method, if the button is not present on the phone (hard button) or not available (softkey) when the URI is processed, the event is discarded.

Example 3-1 Key:Soft1

If the softkey set is changing and disabled while the event is being processed, the request is discarded.

Verify available softkeys by using the QA web pages that the phones web server provides to indicate the active softkey set.

Key

URI Format:

Key:n

Where

n = a Key name.

The following is a complete listing of the Key URIs:

- Key:Line1 to Key:Line34
- Key:KeyPad0 to Key:KeyPad9
- Key:Soft1 to Key:Soft5
- Key:KeyPadStar
- Key:KeyPadPound
- Key:VolDwn
- Key:VolUp
- Key:Headset
- Key:Speaker
- Key:Mute
- Key:NavLeft
- Key:NavRight
- Key:NavSelect
- Key:Info
- Key:Messages
- Key:Services
- Key:Directories
- Key:Settings
- Key:NavUp
- Key:NavDwn
- Key:AppMenu
- Key:Hold

URIs for Invoking Softkey Functionality

You can execute native softkey functionality when the phone executes a Softkey URI. The SoftKey URI allows developers to customize softkey names and layout in the Services and Directories windows while retaining the functionality that the softkeys provide.

Softkey URIs work in menu items and in softkey items in the XML objects for which they natively occur on the phone.

SoftKey

URI Format:

`SoftKey:n`

Where

n = one of the following softkey names:

- Back
- Cancel
- Exit
- Next
- Search
- Select
- Submit
- Update
- Dial
- EditDial
- <<

[Table 3-2](#) contains valid softkey actions for each XSI object type follow. The URI invokes the native functionality that each key possesses in the given object context.

Table 3-2 Valid Softkey Actions for CiscoIPPhoneObject Types

IPPhoneObject	Select	Exit	Update	Submit	Search	<<	Cancel	Next	Dial	Edit Dial
CiscoIPPhoneMenu	X	X								
CiscoIPPhoneIconMenu	X	X								
CiscoIPPhoneText		X	X							
CiscoIPPhoneImage		X	X							
CiscoIPPhoneGraphicMenu		X	X							
CiscoIPPhoneInput				X	X ¹	X	X			
CiscoIPPhoneDirectory							X	X	X ²	X ²

1. Only when used under the Directories button.
2. The SoftKey:Dial and SoftKey:EditDial URIs can be used only for Directory objects, but the Dial:xxx and EditDial:xxx URIs can be used as the URL of any SoftKeyItem or MenuItem. For more details, see the “Miscellaneous URIs” section on page 3-11.

QueryStringParam URI

The QueryStringParam URI allows an application developer to collect more information from the user with less interaction. When the user performs an action with a softkey, you can either append a query string parameter to the URL of the highlighted MenuItem or append the query string parameter from the MenuItem to the URL of the softkey.

URI Format:

```
QueryStringParam:d
```

Where

d = the data to be appended to a corresponding URL.

Example 3-2 QueryStringParam URI in a CiscoIPPhoneMenu object

```
<CiscoIPPhoneMenu>
  <Title>Message List</Title>
  <Prompt>Two Messages</Prompt>
  <MenuItem>
    <Name>Message One</Name>
```

```

        <URL>QueryStringParam:message=1</URL>
    </MenuItem>
    <MenuItem>
        <Name>Message Two</Name>
        <URL>queryStringParam:message=2</URL>
    </MenuItem>
    <SoftKeyItem>
        <Name>Read</Name>
        <URL>http://server/read.asp</URL>
    </SoftkeyItem>
    <SoftKeyItem>
        <Name>Delete</Name>
        <URL>http://server/delete.asp</URL>
    </SoftkeyItem>
</CiscoIPPhoneMenu>

```

Example 3-2 shows how to use the `QueryStringParam` URI in a `CiscoIPPhoneMenu` object. The `CiscoIPPhoneMenu` object includes two `MenuItem`s with `QueryStringParam` URIs. If the user chooses the `MenuItem`(s) with the numeric keypad, the cursor moves to that entry, but nothing executes because the values are `QueryStringParam` URIs.

If the user presses either custom softkey, the currently highlighted `MenuItem` URI value gets appended to the softkey URL that was pressed and requested from the web server.

If you highlight the first `MenuItem` and press the Read softkey, the phone generates the following URL:

```
http://server/read.asp?message=1
```

Example 3-3 *Selecting an Item with Numeric Keypad Calls the URL*

```

<CiscoIPPhoneMenu>
    <Title>Message List</Title>
    <Prompt>Two Messages</Prompt>
    <MenuItem>
        <Name>Messae One</Name>
        <URL>http://server/messages.asp?message=1</URL>
    </MenuItem>
    <MenuItem>
        <Name>Messae Two</Name>
        <URL>http://server/messages.asp?message=2</URL>
    </MenuItem>
    <SoftKeyItem>
        <Name>Read</Name>
        <Position>1</Position><URL>QueryStringParam:action=read</URL>

```

```
</SoftKeyItem>
<SoftKeyItem>
  <Name>Delete</Name>
  <Position>2</Position><URL>QueryStringParam:action=delete</URL>
</SoftKeyItem>
</CiscoIPPhoneMenu>
```

The Cisco Unified IP Phones allow you to implement the `QueryStringParam` URI in either manner although [Example 3-3](#) is not as efficient as [Example 3-2](#). Choose the best way to perform the action based on your applications needs.

[Example 3-3](#) does have a slight advantage in that if the user chooses an item with the numeric keypad, the URL gets called. This would allow you to invoke some default behavior such as to read the message in the example. By highlighting the first message and pressing the Read softkey, the phone creates the following URL: `http://server/messages.asp?message=1&action=read`

Using the `QueryStringParam` URI reduces the size of the XML objects that you generate by not having to repeat redundant portions of a URL in every `MenuItem`.

URIs to Control RTP Streaming

You can invoke RTP streaming via URIs in services. You can instruct the phone to transmit or receive an RTP stream with the following specifications:

- `RTPRx`
- `RTPTx`
- `RTPMRx`
- `RTPMTx`

The supported format of the RTP stream is as follows:

- The codec is G.711 mu-Law.
- The packet size is 20 ms.

The following list gives these possible `CiscoIPPhoneError` codes:

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

Interaction with Call Streaming

- Existing Tx URI streams will be terminated if a new call begins or an existing call is resumed
- Tx URI stream requests received when a call is active will be rejected with an `errorNo=4 unauthorized`. If a call is in a Held state (connected but not actively streaming), the Tx URI request will be accepted, but will be terminated if the call is resumed.



Note Returning `errorNo=4` allows the application to distinguish this error from the normal `errorNo=1 busy` response.

- Existing Rx URI streams will be terminated if a new call begins or an existing call is resumed.

The user has no explicit mechanism for terminating the Rx URI stream independent of the call. Thus, if the Rx stream is not terminated automatically, it would continue to play. For example, a user is listening to Internet radio feed and gets an incoming call. The user answers the call, which either closes or minimizes the Internet radio XSI application. Otherwise, the user has no intuitive way to stop the music stream.

- New Rx URI stream requests received during an active call will be accepted (whisper), but the volume parameter of the URI will be ignored.

If the Rx URI request was done via push, then the associated application is responsible for using push Priority attributes and for stopping and starting the stream.

If the user initiates the Rx URI via an application, then the user likely is not concerned about having the audio mixed with the current call. However, they should also be presented with an option to stop the application, when needed.

- For the Rx URI, the Mute indicator light is only lit when both these conditions are met:
 - There are no active transmit streams from either a call or an XML services stream, and
 - There is at least one active receive stream

For example, if an active call is ended or put on hold while a Rx URI stream is active, the Mute indicator will light.

- If a Rx or Tx URI request is received and there is already an active XML services stream in that direction, then a response with `errorNo=1 Tx/Rx is already active` will be returned. The previous stream must be terminated (either by the user or by an RTP Stop URI) before a new stream can be started.

This response provides visibility to the application if the phone is currently busy. It then allows the application to decide whether or not to terminate the existing stream and start a new one, rather than being controlled by the phone firmware.

RTPRx

The RTPRx URI instructs the phone to receive a Unicast RTP stream or to stop receiving Unicast or Multicast RTP streams.

URI Formats:

```
RTPRx:i:p:v  
RTPRx:Stop
```

Where

i = the IP Address from which the stream is coming.

p = the UDP port on which to receive the RTP stream. Ensure that this is an even port number within the decimal range of 20480 to 32768. If no port is specified, the phone chooses a port and returns it when initiated by a push request.

Stop = the parameter that will stop any active RTP stream from being received on channel one

v = the optional volume setting that controls the volume of stream playback. The supplied value is a percentage of the maximum volume level of the device and must be in the range 0-100. The phone converts the specified percentage into the closest device-supported volume level setting and uses it. After the initial volume level gets set and the stream starts, you can manually change the volume level as needed. If the optional volume parameter does not get included, the current volume setting on the phone gets used as the default.

RTPTx

Use the RTPTx URI to instruct the phone to transmit a Unicast RTP stream or to stop transmitting Unicast or Multicast RTP streams.

URI Formats:

```
RTPTx:i:p  
RTPTx:Stop
```

Where

i = the IP Address to which an RTP stream is transmitted.

p = the UDP port on which to transmit the RTP stream. Ensure that this is an even port number within the decimal range of 20480 to 32768.

Stop = the parameter that will stop any active RTP stream from being transmitted on channel one.

RTPMRx

The RTPMRx URI instructs the phone to receive a Multicast RTP

URI Format:

```
RTPMRx:i:p:v
```

Where

i = the Multicast IP Address from which to receive an RTP stream.

p = the Multicast UDP port from which to receive the RTP stream. Ensure that this is an even port number within the decimal range of 20480 to 32768.

v = the optional volume setting that controls the volume of stream playout. The supplied value is a percentage of the maximum volume level of the device and must be in the range 0-100. The phone converts the specified percentage into the closest device-supported volume level setting and uses it. After the initial volume level gets set and the stream starts, you can manually change the volume level as needed. If the optional volume parameter does not get included, the current volume setting on the phone gets used as the default.

RTPMTx

The RTPMTx URI instructs the phone to transmit a Multicast RTP stream.

URI Formats:

RTPMTx:i:p

Where

i = the Multicast IP Address to which an RTP stream is transmitted.

p = the Multicast UDP port on which to transmit the RTP stream. Ensure that this is an even port number within the decimal range of 20480 to 32768.

Miscellaneous URIs

This section describes the following miscellaneous URIs:

- [Init](#)
- [Dial](#)
- [EditDial](#)
- [Play](#)
- [Display](#)

Init

The Init URI allows an application to initialize a feature or data with the argument that is passed with the URI.

URI Format:

Init:o

Where

o = the Object name.

Valid object name:

CallHistory—When the phone encounters an `Init:CallHistory` URI, it clears the internal call history logs that are stored in the phone. This action initializes Missed Calls, Received Calls, and Placed Calls.

services—When the phone encounters an `Init:Services` URI, it closes the Services application. If Services is not currently open, it has no effect.

Messages—When the phone encounters an `Init:Messages` URI, it closes the Messages application. If Messages is not currently open, it has no effect.

Directories—When the phone encounters an `Init:Directories` URI, it closes the Directories application. If Directories is not currently open, it has no effect.

Dial

The Dial URI initiates a new call to a specified number. The Dial URI invokes when it is contained in a menu item, the menu item is highlighted, and the device is taken off hook.

Activate the Dial URI by one of the following:

- Line button
- Speaker button
- Headset button
- Handset hook switch
- Normal menu item
- Softkey item selection

URI Format:

`Dial:n`

Where

n = the number dialed (such as **Dial:1000**).

EditDial

The EditDial URI initiates a new call to a specified number. The EditDial URI invokes when it is contained in a menu item and the menu item is highlighted.

Activate the EditDial URI by one of the following:

- Line button
- Speaker button
- Headset button
- Handset hook switch
- Normal menu item
- Softkey item selection

URI Format:

`EditDial:n`

Where

n = the number dialed (such as **EditDial:1000**).

Play

The Play URI downloads an audio file from the TFTP server and plays through the phone speaker. This same mechanism also plays ring files, and the format of the files is the same. You could use the Play URI to play files that are in the Ringlist.xml or those that are not. If the phone is equipped with an MWI light, it will be flashing while the audio file is playing, providing a visual alert as well.



Note

The Play URI is a synchronous request. If the request is pushed to the phone via HTTP, the HTTP response (CiscoIPPhoneResponse object) is not returned until after the playback has completed.

Interaction with Incoming Calls

The Play URI and incoming calls (ringing) have equal priority access to the DSP ringer resources resulting in the following interactions:

- If a Play URI is currently playing, an incoming call (ringing) will not preempt the Play URI; the Play URI will finish playing first.
- If the phone is ringing and a Play URI request is sent to the phone, the execution of the Play URI defers until the phone stops ringing (the DSP ringer resource becomes available) and then the Play URI will play.

URI Format:

`Play:f`

Where

f = the filename of a raw audio file in the TFTP path (such as **Play:Classic2.raw**).

The audio files for the rings must meet the following requirements for proper playback on Cisco Unified IP Phones:

- Raw PCM (no header)
- 8000 samples per second
- 8 bits per sample
- uLaw compression
- Maximum ring size—16080 samples
- Minimum ring size—240 samples
- Number of samples in the ring is evenly divisible by 240.
- Ring starts and ends at the zero crossing.

To create PCM files for custom phone rings, you can use any standard audio editing packages that support these file format requirements.

Display

The Display URI is available only on those Cisco Unified IP Phones that have a color backlight on the phone display, including the Cisco Unified IP Phone 7970G and 7971G-GE. Using the Display URI, you can control how long the backlight remains on or off.

Note, however, that other administrator-controlled or user-indicated display settings take precedence over the Display URI. As such, various phone states (such as phone startup, incoming and active calls, or other user input states) override the Display URI settings.

URI Format:

`Display:State:Interval`

Where

State = whether the phone display is turned on (**on**) or off (**off**) or set to default (**default**) to return the display to its specified state.

Interval = duration (in minutes) in which the the phone state remains in the specified state (unless activated by automated or user input). Value must be an integer ranging from 0-1440 minutes. If the value is set to 0, the display remains in the indicated state indefinitely (unless activated by automated or user input).

For example:

- `Display:Off:60` turns the phone display off for 1 hour (60 minutes).
- `Display:On:10` turns the phone display on for 10 minutes.
- `Display:Off:0` turns off the display off until activated.
- `Display:Default` returns the display to its specified state for that time.

Vibrate

The Vibrate URI is available on the Cisco Unified IP Phones 7920G and 7921G wireless phone models, and it enables third-party applications to invoke the phone's vibration capabilities for silent alerts, similar to the way in which the Play URI plays audible alerts. If the Vibrate parameters are not specified or if the device is unable to support custom Vibrate sequences, the device will execute its default vibrate sequence.

URI Format:

```
Vibrate:vibrateDuration:silenceDuration:count
```

Where

vibrateDuration = duration (in milliseconds) in which the vibrate state remains on. Value must be an integer ranging from 0-65536 milliseconds.

silenceDuration = duration (in milliseconds) in which the the vibrate state remains off. Value must be an integer ranging from 0-65536 milliseconds.

count = number of times to repeat the vibrate on and off sequence.

For example:

- `Vibrate:1000:0:1` initiates a single vibrate for 1 second.
- `Vibrate:500:1500:5` initiates five vibrations each lasting for 500 ms. followed by 1500 ms of silence.

