



Internal URI Features

- [Internal URI Overview, on page 1](#)
- [Supported URIs by Phone Model, on page 1](#)
- [Device Control URIs, on page 5](#)
- [XML Displayable Object URIs, on page 15](#)
- [Multimedia URIs, on page 20](#)
- [Telephony URIs, on page 27](#)
- [Application Management URIs, on page 30](#)

Internal URI Overview

Internal uniform resource identifiers (URIs) provide access to embedded phone features such as placing calls, playing audio files, and invoking built-in object features.

Supported URIs by Phone Model

The following tables list the URIs. The notes mentioned in the tables follow the final table. For SPA and Multiplatform phones, see the final table.

Table 1: URIs Supported by Cisco IP Phone Services SDK for Cisco IP Phone 7800 Series (On-Premise)

URI	7811, 7821, 7841, 7861 On-Premise	7832 On-Premise
Key (See Note 6)	Supported	Supported
Softkey	Supported	Supported
Init	Supported	Supported
Dial, EditDial	Supported	Supported
Play	Supported	Supported

URI	7811, 7821, 7841, 7861 On-Premise	7832 On-Premise
QueryStringParam	Supported	Supported
Unicast RTP	Supported	Supported
Multicast RTP	Supported	Supported
Display	Not supported	Not supported
Vibrate	Not supported	Not supported
Notify (See Note 2)	Not supported	Not supported
SendDigits (See Note 2)	Not supported	Not supported
Application (See Note 2)	Not supported	Not supported
Device	Not supported	Not supported

Table 2: URIs Supported by Cisco Unified IP Phone Services SDK for Cisco IP Phone 8800 Series and Cisco IP Conference Phones (On-Premise)

URI	8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR, 8875, 8875NR	8831	8832
Key	Supported	Supported	Supported
Softkey	Supported (Except for 8875 and 8875NR)	Supported	Supported
Init	Supported	Supported	Supported
Dial, EditDial	Supported	Supported	Supported
Play	Supported	Supported	Supported
QueryStringParam	Supported	Supported	Supported
Unicast RTP	Supported	Supported	Supported
Multicast RTP	Supported	Supported	Supported
Display	Supported	Supported	Supported
Vibrate	Not supported	Not supported	Not supported
Notify (See Note 2)	Supported	Supported	Supported

URI	8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR, 8875, 8875NR	8831	8832
SendDigits (See Note 2)	Supported	Supported	Supported
Application (See Note 2)	Supported	Not supported	Supported
Device	Supported	Not supported	Supported

Table 3: URIs Supported by Cisco Unified IP Phone Services SDK for Cisco Wireless IP Phone 8820 Series and Cisco Wireless Phone 800 Series

URI	8821	840/860
Key	Supported	Not Supported
Softkey	Supported	Supported
Init	Supported	Not Supported
Dial, EditDial	Supported (See Note 1)	Not Supported
Play	Supported	Not Supported
QueryStringParam	Supported	Not Supported
Unicast RTP	Supported	Supported
Multicast RTP	Supported	Supported
Display	Supported	Not Supported
Vibrate	Supported	Not Supported
Notify (See Note 2)	Not supported	Not Supported
SendDigits (See Note 2)	Not supported	Not Supported
Application (See Note 2)	Not supported	Not Supported
Device	Supported	Not Supported

Table 4: URIs Supported by Multiplatform Phones

URI	6800 Series	7800 Series	7832	8800 Series
Key (See Note 3 and Note 6)	Supported	Supported	Supported	Supported

Supported URIs by Phone Model

URI	6800 Series	7800 Series	7832	8800 Series
Softkey	Supported	Supported	Supported	Supported
Init (See Note 4)	Supported	Supported	Supported	Supported
Dial	Supported	Supported	Supported	Supported
EditDial	Supported	Supported	Supported	Supported
Play	Supported (See Note 5)	Supported (See Note 5)	Supported (See Note 5)	Supported (See Note 5)
QueryStringParam	Not supported	Not supported	Not supported	Not supported
Unicast RTP	Not supported	Not supported	Not supported	Not supported
Multicast RTP	Not supported	Not supported	Not supported	Not supported
Display	Not supported	Not supported	Not supported	Not supported
Vibrate	Not supported	Not supported	Not supported	Not supported
Notify	Not supported	Not supported	Not supported	Not supported
SendDigits	Not supported	Not supported	Not supported	Not supported
Application	Not supported	Not supported	Not supported	Not supported
Device	Supported	Supported	Supported	Supported

Table 5: URIs Supported by Cisco Desk Phone 9800 Series

URI	9841 and 9851
Key	Supported
Softkey	Supported
Init	Supported
Dial, EditDial	Supported
Play	Supported
QueryStringParam	Supported
Unicast RTP	Supported
Multicast RTP	Supported
Display	Supported
Vibrate	Not supported

URI	9841 and 9851
Notify	Supported
SendDigits	Supported
Application	Supported
Device	Supported



- Note**
1. Cisco Wireless IP Phone 8821 support only the Dial:N format.
 2. Requires Cisco Unified IP Phone firmware version 8.3(2) or later, which contains an updated XML parser.
 3. Only Key:Headset supported.
 4. Only Init:Call History supported.
 5. Multiplatform phones support Play, but only for ringtones.
 6. The Cisco IP Phone 7800 Series uses Key:Hold.
The Cisco IP Phone 8800 Series uses Key:FixedFeature3 instead of Key:Hold.

Device Control URIs

These sections describe the device control URIs.

Key

The *Key* URI allows a program 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.

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

The following tables list the *Key* URIs and the phone models in which these softkeys are supported.

Supported Key URIs for 7800 Series

Table 6: Key URIs for Cisco IP Phone 7800 Series (Unified CM)

Key URIs	7811, 7821, 7841, 7861	7832
Key:Applications	Supported	Not supported
Key:AppMenu	Not supported	Not supported
Key:Contacts	Not supported	Not supported

Key URIs	7811, 7821, 7841, 7861	7832
Key:Directories	Supported	Not supported
Key:Feature1 to Key:Feature120	Not supported	Not supported
Key:FixedFeature1 to Key:FixedFeature3	Not supported	Not supported
Key:Headset	Supported	Not supported
Key:Hold	Supported	Not supported
Key:Info	Supported	Not supported
Key:KEMPage	Not supported	Not supported
Key:KeyPad 0 to Key:KeyPad9	Supported	Supported
Key:KeyPadPound	Supported	Supported
Key:KeyPadStar	Supported	Supported
Key:Line1 to Key:Line120	Supported	Not supported
Key:Messages	Supported	Not supported
Key:Mute	Supported	Supported
Key:NavBack	Not supported	Not supported
Key:NavDwn	Supported	Supported
Key:NavLeft	Not supported	Not supported
Key:NavRight	Not supported	Not supported
Key:NavSelect	Supported	Supported
Key:NavUp	Supported	Supported
Key:NavOffhook	Not supported	Not supported
Key:NavOnhook	Not supported	Not supported
Key:PTT	Not supported	Not supported
Key:Release	Not supported	Not supported
Key:Services	Supported	Not supported
Key:Session1 to Session6	Not supported	Not supported
Key:Settings	Supported	Not supported
Key: Soft1 to Soft4	Supported	Supported

Key URIs	7811, 7821, 7841, 7861	7832
Key:Speaker	Supported	Supported
Key:VolDown	Supported	Supported
Key:VolUp	Supported	Supported



Note The Cisco IP Phone 7811 supports 1 line. It does not have feature buttons.

Supported Key URIs for 8800 Series

Table 7: Key URIs for Cisco IP Phone 8800 Series and Cisco IP Conference Phone 8830 Series (Unified CM)

Key URIs	8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR	8875, 8875NR	8831	8832
Key:Applications	Supported	Supported	Not supported	Not supported
Key:AppMenu	Not supported	Not supported	Not supported	Not supported
Key:Contacts	Supported	Supported	Not supported	Not supported
Key:Directories	Supported	Supported	Not supported	Not supported
Key:Feature1 to Key:Feature120	Supported	Not supported	Not supported	Not supported
Key:FixedFeature1 to 3	Supported	Not supported	Not supported	Not supported
Key:Headset	Supported	Supported	Not supported	Not supported
Key:Hold	Supported	Supported	Not supported	Not supported
Key:Info	Not supported	Not supported	Not supported	Not supported
Key:KemPage	Supported	Not supported	Not supported	Not supported
Key:KeyPad0 to Key:KeyPad9	Supported	Supported	Supported	Supported
Key:KeyPadPound	Supported	Supported	Supported	Supported
Key:KeyPadStar	Supported	Supported	Supported	Supported
Key:Line1 to Key:Line120	Supported	Not supported	Not supported	Not supported
Key:Messages	Supported	Supported	Not supported	Not supported

Key URIs	8811, 8841, 8845, 8851, 8851NR, 8861, 8865, 8865NR	8875, 8875NR	8831	8832
Key:Mute	Supported	Supported	Not supported	Not supported
Key:NavBack	Not supported	Not supported	Not supported	Not supported
Key:NavDwn	Supported	Not supported	Not supported	Supported
Key:NavLeft	Supported	Not supported	Not supported	Not supported
Key:NavRight	Supported	Not supported	Not supported	Not supported
Key:NavSelect	Supported	Not supported	Not supported	Supported
Key:NavUp	Supported	Not supported	Not supported	Supported
Key:Offhook	Not supported	Not supported	Not supported	Not supported
Key:Onhook	Not supported	Not supported	Not supported	Not supported
Key:PTT	Not supported	Not supported	Not supported	Not supported
Key:Release	Supported	Supported	Not supported	Not supported
Key:Services	Supported	Supported	Not supported	Not supported
Key:Session1 to Key:Session6	Supported	Not supported	Not supported	Not supported
Key:Settings	Supported	Not supported	Not supported	Not supported
Key:Soft1 to Key:Soft5	Supported	Not supported	Supported	Supported
Key:Speaker	Supported	Supported	Supported	Supported
Key:VolDwn	Supported	Supported	Supported	Supported
Key:VolUp	Supported	Supported	Supported	Supported

**Note**

- The Cisco IP Phone 8811, 8841, and 8845 support 5 lines, 5 sessions, and 5 features. The Cisco IP Phones 8851 and 8851NR support 77 lines, 5 sessions, and 77 features. The Cisco IP Phone 8861, 8865, and 8865NR supports 113 lines, 5 sessions, and 113 features.
- Key:Directories is supported on the Cisco IP Phone 8800 series starting with Firmware Release 11.0
- Cisco Video Phone 8875 and 8875NR don't have the Navigation keys and don't support the navigation key URIs.

Supported Key URIs for 9800 Series

Table 8: Key URIs for Cisco Desk Phone 9800 Series (Unified CM)

Key URIs	9841 and 9851
Key:Applications	Supported
Key:AppMenu	Not supported
Key:Contacts	Supported
Key:Directories	Supported
Key:Feature1 to Key:Feature120	Supported
Key:FixedFeature1 to Key:FixedFeature3	Supported
Key:Headset	Supported
Key:Hold	Supported
Key:Info	Not supported
Key:KEMPage	Not supported
Key:KeyPad 0 to Key:KeyPad9	Supported
Key:KeyPadPound	Supported
Key:KeyPadStar	Supported
Key:Line1 to Key:Line120	Supported
Key:Messages	Not supported
Key:Mute	Supported
Key:NavBack	Not supported
Key:NavDwn	Supported
Key:NavLeft	Supported
Key:NavRight	Supported
Key:NavSelect	Supported
Key:NavUp	Supported
Key:NavOffhook	Not supported
Key:NavOnhook	Not supported
Key:PTT	Not supported
Key:Release	Supported

Supported Key URIs for Multiplatform Phones

Key URIs	9841 and 9851
Key:Services	Supported
Key:Session1 to Session6	Not supported
Key:Settings	Supported
Key: Soft1 to Soft4	Supported
Key:Speaker	Supported
Key:VolDown	Supported
Key:VolUp	Supported

Supported Key URIs for Multiplatform Phones

Table 9: Key URIs Supported by Multiplatform Phones

Key URIs	6800 Series	7800 Series	7832	8800 Series
Key:Applications	Not supported	Not supported	Not supported	Not supported
Key:AppMenu	Not supported	Not supported	Not supported	Not supported
Key:Contacts	Not supported	Not supported	Not supported	Not supported
Key:Directories	Supported	Supported	Supported	Supported
Key:Feature1 to Key:Feature120	Not supported	Not supported	Not supported	Not supported
Key:FixedFeature1 to 3	Not supported	Not supported	Not supported	Not supported
Key:Headset	Supported	Supported	Not supported	Supported
Key:Hold	Supported	Supported	Supported	Supported
Key:Info	Supported	Supported	Supported	Supported
Key:KemPage	Not supported	Not supported	Not supported	Not supported
Key:KeyPad0 to Key:KeyPad9	Supported	Supported	Supported	Supported
Key:KeyPadPound	Supported	Supported	Supported	Supported
Key:KeyPadStar	Supported	Supported	Supported	Supported
Key:Line1 to Key:Line120	Supported	Supported	Not supported	Supported
Key:Messages	Supported	Supported	Not supported	Supported

Key URIs	6800 Series	7800 Series	7832	8800 Series
Key:Mute	Supported	Supported	Not supported	Supported
Key:NavBack	Not supported	Not supported	Not supported	Supported (Except 8875)
Key:NavDwn	Supported	Supported	Supported	Supported (Except 8875)
Key:NavLeft	Not supported	Not supported	Not supported	Supported (Except 8875)
Key:NavRight	Not supported	Not supported	Not supported	Supported (Except 8875)
Key:NavSelect	Supported	Supported	Supported	Supported (Except 8875)
Key:NavUp	Supported	Supported	Supported	Supported (Except 8875)
Key:Offhook	Not supported	Not supported	Not supported	Not supported
Key:Onhook	Not supported	Not supported	Not supported	Not supported
Key:PTT	Not supported	Not supported	Not supported	Not supported
Key:Release	Not supported	Not supported	Not supported	Not supported
Key:Services	Not supported	Not supported	Not supported	Not supported
Key:Session1 to Key:Session6	Not supported	Not supported	Not supported	Not supported
Key:Settings	Not supported	Not supported	Not supported	Not supported
Key:Soft1 to Key:Soft4	Supported	Supported	Supported	Supported
Key:Speaker	Supported	Supported	Not supported	Supported
Key:VolDwn	Supported	Supported	Supported	Supported
Key:VolUp	Supported	Supported	Supported	Supported

**Note**

- The Cisco IP Phone 7811 supports 1 line. It does not have feature buttons.
- Cisco Multiplatform Phones support four softkeys.

Supported Key URIs for Wireless Phones

Table 10: Key URIs with Supported by Cisco Wireless Phone 8820 Series

Key URIs	8821
Key:Applications	Not supported
Key:AppMenu	Not supported
Key:Contacts	Not supported
Key:Directories	Not supported
Key:Feature1 to Key:Feature120	Not supported
Key:FixedFeature1 to 3	Not supported
Key:Headset	Not supported
Key:Hold	Supported
Key:Info	Supported
Key:KemPage	Not supported
Key:KeyPad0 to Key:KeyPad9	Supported
Key:KeyPadPound	Supported
Key:KeyPadStar	Supported
Key:Line1 to Key:Line120	Not supported
Key:Messages	Not supported
Key:Mute	Supported
Key:NavBack	Not supported
Key:NavDwn	Supported
Key:NavLeft	Supported
Key:NavRight	Supported
Key:NavSelect	Supported
Key:NavUp	Supported
Key:Offhook	Supported
Key:Onhook	Supported
Key:PTT	Supported
Key:Release	Not supported

Key URIs	8821
Key:Services	Not supported
Key:Session1 to Key:Session6	Not supported
Key:Settings	Not supported
Key:Soft1 to Key:Soft2	Supported
Key:Speaker	Supported
Key:VolDwn	Supported
Key:VolUp	Supported

Key URI Format

Key:n

Where

n = a Key name

Display

The *Display* URI is available only on those Cisco Unified IP Phones that have a color backlight on the phone display. 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. Therefore, various phone states (such as phone startup, incoming and active calls, or other user input states) override the *Display* URI settings.

Display URI Format

Display:State:Interval

Where

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

Interval = duration (in minutes) in which 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).

Examples

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.

Settings Menu

With Settings Menu URIs, you can remotely interact with the **Settings** menu on the phone.



Note Settings Menu URIs are available only on Cisco Video Phone 8875 and 8875NR deployed on Cisco Unified Communications Manager.

URI Format

Cisco Video Phone 8875 supports three types of URIs: SettingsMenu, Button, and Key.

SettingsMenu URI Format

SettingsMenu:<menu name>

SettingsMenu URIs enable you to remotely open the menu and submenu items in the **Settings** menu. Ensure that the <menu name> matches the exact display name of the corresponding menu item on the phone screen. Before accessing a menu, open the **Settings** menu with the URI `Key:Applications`.

Here are some examples:

- `SettingsMenu:Bluetooth`— Open the **Bluetooth** menu when in the **Settings** menu
- `SettingsMenu>About this device`—
Open the **About this device** menu when in the **Settings** menu
- `SettingsMenu:Network settings`—
Open the **Network settings** menu when in the **Network connection** menu
- `SettingsMenu:Alternate TFTP`— Toggle on or off the **Alternate TFTP** option when in the **Network settings** menu
- `SettingsMenu:TFTP server1`— Put the focus in the **TFTP server1** field when in the **Network settings** menu

Button URI Format

Button:<button name>

Button URIs allow you to remotely interact with the soft buttons in the **Settings** menu.

Here are some examples:

- `Button:Back`— Go back to the previous menu page
- `Button:Backspace`— Delete the input by one character to the left
- `Button:Apply/Submit`— Submit the changes that you've made, such as TFTP server address

Key URI Format

Key:<key name>

The Key URI enables you to remotely enter digits and the characters *****, **.**, and **#** in the phone menu.

Here are some examples:

- Key:Key_0— Enter the digit 0
- Key:Key_6— Enter the digit 6
- Key:Key_Asterisk— Enters the star character (*)
- Key:Key_Period— Enters a dot (.) for IP addresses
- Key:Key_NumberSign— Enters the pound character (#)

Example

This example is for remotely entering the alternate TFTP server address in the phone menu.

1. Open the **Settings** menu with the following command:

```
<CiscoIPPhoneExecute>
  <ExecuteItem Priority="o"> URL="Key:Applications"/>
</CiscoIPPhoneExecute>
```

2. Run the following commands to open the **Network connection > Network settings** menu, toggle on Alternate TFTP, enter the TFTP server address, and apply the setting.

For example, the TFTP server address is **100.7.34.1**.

```
<CiscoIPPhoneExecute>
  <ExecuteItem Priority="o"> URL="SettingsMenu:Network connection"/>
  <ExecuteItem Priority="o"> URL="SettingsMenu:Network settings"/>
  <ExecuteItem Priority="o"> URL="SettingsMenu:Alternate TFTP"/>
  <ExecuteItem Priority="o"> URL="SettingsMenu:TFTP server1"/>
  <ExecuteItem Priority="o"> URL="Key:Key_1"/>
  <ExecuteItem Priority="o"> URL="Key:Key_Period"/>
  <ExecuteItem Priority="o"> URL="Key:Key_0"/>
  <ExecuteItem Priority="o"> URL="Key:Key_0"/>
  <ExecuteItem Priority="o"> URL="Key:Key_Period"/>
  <ExecuteItem Priority="o"> URL="Key:Key_7"/>
  <ExecuteItem Priority="o"> URL="Key:Key_Period"/>
  <ExecuteItem Priority="o"> URL="Key:Key_3"/>
  <ExecuteItem Priority="o"> URL="Key:Key_4"/>
  <ExecuteItem Priority="o"> URL="Key:Key_Period"/>
  <ExecuteItem Priority="o"> URL="Key:Key_1"/>
  <ExecuteItem Priority="o"> URL="Button:Apply"/>
</CiscoIPPhoneExecute>
```

If you have entered a wrong number, run the following command to delete your input by one character to the left:

```
<CiscoIPPhoneExecute>
  <ExecuteItem Priority="o"> URL="Button:Backspace"/>
</CiscoIPPhoneExecute>
```

3. Run the following command repetitively until you exit the **Settings** menu.

```
<CiscoIPPhoneExecute>
  <ExecuteItem Priority="o"> URL="Button:Back"/>
</CiscoIPPhoneExecute>
```

XML Displayable Object URIs

These sections describe the XML displayable object URIs.

SoftKey

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.



Note The Softkey URI is not supported in the Execute object.

SoftKey URI Format

SoftKey:n

Where

n = one of the following softkey names:

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

The following table 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 11: Valid Softkey Actions for CiscoIPPhoneObject Types (Part 1)

IPPhoneObject (see note 1)	Back (See note 2)	Select	Exit (See note 2)	Update	Submit
CiscoIPPhoneMenu	X	X	X		
CiscoIPPhoneIconMenu	X	X	X		
CiscoIPPhoneText	X		X	X	

IPPhoneObject (see note 1)	Back (See note 2)	Select	Exit (See note 2)	Update	Submit
CiscoIPPhoneImage	X		X	X	
CiscoIPPhoneGrapphicMenu	X		X	X	
CiscoIPPhoneInput	X				X
CiscoIPPPhoneDirectory	X				

Table 12: Valid Softkey Actions for CiscoIPPhoneObject Types (Part 2)

IPPhoneObject (see note 1)	Search	<<	Cancel	Next	Dial	EditDial
CiscoIPPhoneMenu						
CiscoIPPhoneIconMenu						
CiscoIPPhoneText						
CiscoIPPhoneImage						
CiscoIPPhoneGrapphicMenu						
CiscoIPPhoneInput	X (see note 3)	X	X			
CiscoIPPPhoneDirectory			X	X	X (see note 4)	X (see note 4)



- Note**
1. The *SoftKey* URI is not allowed in an Execute object, except for the Cisco Wireless IP Phone 8821 and 8821-EX.
 2. For the Cisco Wireless IP Phone 8821 and 8821-EX Firmware Release 11.0(4) and previous, the *Softkey:Back* button takes the user up one screen level and the *Softkey:Exit* closes all open applications. For the Cisco Wireless IP Phone 8821 and 8821-EX Firmware Release 11.0(4)SR1 and later, the *Softkey:Back* and *Softkey:Exit* buttons take the user up one screen level.
 3. Only when used under the Directories button.
 4. 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*.

The following table shows this table from the document and denotes the Cisco Wireless Phone 840 or 860 behavior:

Table 13: Valid Softkey Actions for CiscoIPPhoneObject Types (Part 1)

IPPhoneObject	Select	Select	Exit	Exit	Update	Update	Submit	Submit
	8821	840/860	8821	840/860	8821	840/860	8821	840/860
CiscoIPPhoneMenu	X	X	X	X				
CiscoIPPhoneIconMenu	X	N/A	X	N/A				
CiscoIPPhoneText			X	X	X	X		
CiscoIPPhoneImage			X	N/A	X	N/A		
CiscoIPPhoneGraphicMenu				X		X		
CiscoIPPhoneInput			X	N/A	X	N/A		
CiscoIPPPhoneDirectory							X	N/A

Table 14: Valid Softkey Actions for CiscoIPPhoneObject Types (Part 2)

IPPhoneObject (see note 1)	Search	Search	<<	<<	Cancel	Cancel	Next	Next
	8821	840/860	8821	840/860	8821	840/860	8821	840/860
CiscoIPPhoneMenu								
CiscoIPPhoneIconMenu								
CiscoIPPhoneText								
CiscoIPPhoneImage								
CiscoIPPhoneGraphicMenu								
CiscoIPPhoneInput	X	N/A	X	N/A	X	N/A		
CiscoIPPPhoneDirectory					X	N/A	X	N/A

Table 15: Valid Softkey Actions for CiscoIPPhoneObject Types (Part 3)

IPPhoneObject (see note 1)	Dial	Dial	EditDial	EditDial
	8821	840/860	8821	840/860
CiscoIPPhoneMenu				
CiscoIPPhoneIconMenu				
CiscoIPPhoneText				
CiscoIPPhoneImage				
CiscoIPPhoneGraphicMenu				

IPPhoneObject (see note 1)	Dial 8821	Dial 840/860	EditDial 8821	EditDial 840/860
CiscoIPPhoneInput				
CiscoIPPhoneDirectory	X	N/A	X	N/A



Note Cisco Wireless Phone 840/860 support for CiscoIPPhoneInput and CiscoIPPhoneDirectory will be added in future releases.

Related Topics

[Telephony URIs](#), on page 27

QueryStringParam

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.

QueryStringParam URI Format

QueryStringParam:d

Where

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

Example: QueryStringParam URI in 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: Item Selection with Numeric Keypad Calls URI

The following example 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 the user highlights the first *MenuItem* and press the Read softkey, the phone generates the following URL:

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

```
<CiscoIPPhoneMenu>
<Title>Message List</Title>
<Prompt>Two Messages</Prompt>
<MenuItem>
  <Name>Message One</Name>
  <URL>http://server/messages.asp?message=1</URL>
</MenuItem>
<MenuItem>
  <Name>Message 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>
```

QueryStringParam URI Example Discussion

The Cisco Unified IP Phones allow you to implement the *QueryStringParam* URI in either manner although [Example: Item Selection with Numeric Keypad Calls URI, on page 20](#) is not as efficient as [Example: QueryStringParam URI in CiscoIPPhoneMenu Object, on page 19](#). Choose the best way to perform the action based on your applications needs.

The Item selection example has a slight advantage in that if the user chooses an item with the numeric keypad, the URL gets called. This action 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 removing redundant portions of a URL in every *MenuItem*.

Multimedia URIs

These sections describe the multimedia URIs.

RTP Streaming

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

- RTPRx
- RTPTx
- RTPMRx
- RTPMTx



Note For some Cisco Unified IP Phone models, the RTP Streaming URIs have been deprecated by the RTP Streaming API.

The supported format of the RTP stream is:

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

The possible CiscoIPPhoneError codes are:

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

Related Topics

[RTP Streaming API](#)

Interaction with Call Streaming

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



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

- Existing Rx or MRx URI streams are terminated if a new call begins or an existing call resumes.

The user has no explicit mechanism for terminating the Rx or MRx URI stream independent of the call. Thus, if the Rx or MRx 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 or MRx URI stream requests received during an active call are accepted (whisper), but the volume parameter of the URI is ignored.

If the Rx or MRx URI request was done using 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 or MRx URI using an application, then the user likely is not concerned about having the audio mixed with the current call. However, the user should also be presented with an option to stop the application, when needed.

- For the Rx or MRx 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.
 - There is at least one active receive stream.

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

- If a Rx or MRx or Tx or MTx 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 is 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.

RTPRx 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 play out. 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.

RTPTx 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 received on channel one

RTPMRx

The *RTPMRx* URI instructs the phone to receive a Multicast RTP.

RTPMRx URI Format

RTPMRx:i:p:v

Where

i = the Multicast IP Address from which to receive an RTP stream. For information on selecting a Multicast IP Address, see the *Cisco Unified Communications System SRND*, the *IANA guidelines*, and your local network administration policies.

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 play out. 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.

RTPMTx URI Formats

RTPMTx:i:p

Where

i = the Multicast IP Address to which an RTP stream is transmitted. For information on selecting a Multicast IP Address, see the *Cisco Unified Communications System SRND*, the *IANA guidelines*, and your local network administration policies.

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.

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 a message waiting light, the light will flash 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 using HTTP, the HTTP response (*CiscoIPPhoneResponse* object) is not returned until after the playback has completed.

Play URI 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.

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.

XSI Audio Path Control

The XSI Audio Path Control feature enables XSI calls to specify if the audio is played on the speakerphone or handset speaker of the phone.



Note The feature is available on Cisco Wireless IP Phone 8821.

The XSI Audio Path Control feature utilizes the RTP URI which has been extended to give the administrator this option to specify whether audio received via XSI is played through the speaker phone or handset speaker of the Cisco IP Phone.

RTP URI Format

RTPRx:i:p:v:s or RTPMRx:i:p:v:s

Where

i = equals IP address (x.x.x.x).

p = equals UDP port (20480-32768).

v = volume (0-100).

s = specifies where the audio for an XSI call should be played.

- If *s* = 0, then the audio for the XSI call will be played to the speaker phone.
- If *s* = 1, then the audio for the XSI call will be played to the handset speaker or headset.
- If *s* = 2, then the audio for the XSI call will be played to the current audio path.
- If *s* is not present, then the audio for the XSI call is played to the speaker phone.

Examples

XSI Audio Path	Stream Type	RTP URI Example
Speakerphone	Unicast	RTPRx:10.0.0.10:20500 RTPRx:10.0.0.10:20500::0 RTPRx:10.0.0.10:20500:100:0
Handset/Headset	Unicast	RTPRx:10.0.0.10:20500::1 RTPRx:10.0.0.10:20500:100:1
Speakerphone	Multicast	RTPMRx:10.0.0.10:20500 RTPMRx:10.0.0.10:20500::0 RTPMRx:10.0.0.10:20500:100:0
Handset/Headset	Multicast	RTPMRx:10.0.0.10:20500::1 RTPMRx:10.0.0.10:20500:100:1

Device

The *Device* URI instructs the device to automatically unlock the input or display interface without the user unlocking the device manually.

The *Device* URI accepts these commands:

- (All phones) Unlock: If the device is configured to automatically lock the input or display interface, the normal idle timeout behavior applies and the device is automatically locked again.
- (8821 only) GeneratePRT and PRTStatus: See [Create a Remote Problem Report with CiscoIPPhoneExecute](#).

Device URI Format

Device:{*command*}

Where

command = The command the device follows:

- Type: Enum
- Valid Value: Unlock
- Default-value: N/A

Device URI Example

This alert example performs the following actions:

1. Plays a tone on the phone
2. Unlocks the phone
3. Displays an alarm message on the phone

```
<CiscoIPPhoneExecute>
  <ExecuteItem URL="Device:Unlock"/>
  <ExecuteItem URL="Play:alert.wav"/>
</CiscoIPPhoneExecute>
```

On processing the above command, the following response is sent:

```
<CiscoIPPhoneText>
<Title>Alert</Title>
<Prompt>Urgent</Prompt>
<Text>
Please go to room 1234.
</Text>
<SoftKeyItem>
<Name>Accept</Name>
<URL>http://<ip>/AlertResponse.jsp?reason=accept</URL>
</SoftKeyItem>
<SoftKeyItem>
<Name>Busy</Name>
<URL>http://<ip>/AlertResponse.jsp?reason=busy</URL>
</SoftKeyItem>
</CiscoIPPhoneText>
```

Device URI Error and Response

When the *Device* URI is invoked from an Execute object, it uses the standard URI Status and Data values in the *ResponseItems*.

Condition	Status	Data
Executed successfully	0 (Success)	Success
URI syntax is invalid	1 (Parse error)	Invalid URI
URI is not supported	6 (Internal error)	URI not found

Telephony URIs

These sections describe the telephony URIs.

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 methods:

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

Dial URI Format

Dial:{*dialSequence*}[:{*useAppUI*}:{*applicationId*}[:{*audibleFeedback*}]

Where

dialSequence = The sequence of DTMF digits to be dialed. Commas represent 1 second pauses.

- Value Type: String
- Values: minLength=0, no maxLength, can only contain 0123456789#*ABCD and comma (,)
- Default value: N/A

useAppUI = Specifies whether or not this application will be used as the user interface for this call. A value of true will cause the application to keep UI focus when the call is made instead of switching to the Call UI application. The *appId* must be specified or this parameter will have no effect: it will always be false. This optional field is supported only on the Cisco IP Phone 8800 Series.

- Value Type: boolean
- Values: 0 or 1 (0=false 1=true)
- Default value: 0

applicationId = The unique name of the XSI web application requesting this call. This optional field is supported only on the Cisco Unified IP Phone 7900, 8800, 8900, and 9900 Series.

- Value Type: String
- Values: minLength=1, no maxLength, cannot contain semicolons – should be in the format Company/Product.
- Default value: Nil, which means this dial request will not be associated with any application

audibleFeedback = Whether or not to provide audible feedback to the user when the DTMF digits are dialed. This optional field is supported only on the Cisco Unified IP Phone 8800 Series.

- Value Type: Boolean
- Values: 0, 1 (0=false 1=true)
- Default value: 1

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 methods:

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

EditDial URI Format

EditDial:n

Where

n = the number dialed

Example

EditDial:1000 initiates a call to the phone with DN 1000.

SendDigits

The *SendDigits* URI instructs the phone to send a specified sequence of DTMF digits in-band within the media stream of the current active (streaming) call.

Audible feedback to the user can be enabled or disabled and an optional application ID can be specified to ensure that the DTMF digits will only be sent to the call which is associated with a specific application.

SendDigits URI Format

SendDigits:dtmfSequence:audibleFeedback::applicationId

Where

dtmfSequence = the sequence of DTMF digits to be sent. Value must contain only 0123456789#*ABCD and comma (.). The comma represents a one second pause.

audibleFeedback = indicates whether to provide audible feedback to the user as the DTMF digits are entered. Values can be 0 (false) or 1 (true).

applicationId = optional identifier of the application associated with the call which must receive the DTMF digits. Value must be 0-64 and cannot contain colons. The default value is null indicating that the active call should receive the DTMF digits, regardless of any application association.

Example

Make a call using a calling card service that implements these steps:

1. Connects to a 800 calling card service (using the *Dial* URI).
2. Application waits to give call time to connect.
3. Dials the destination number, ensuring that the digits can only be dialed from this application.
4. Pauses 2 seconds.
5. Dials the calling card number.
6. Pauses 1 second.
7. Dials the pin number.

```
<CiscoIPPhoneExecute>
  <ExecuteItem URL="Dial:918005551212:1:Cisco/Dialer"/>
</CiscoIPPhoneExecute>
<CiscoIPPhoneExecute>
  <ExecuteItem URL="SendDigits:6185551212,,987654321,1234:1:Cisco/Dialer"/>
</CiscoIPPhoneExecute>
```

SendDigits Error and Response

When the *SendDigits* URI is invoked from an *Execute* object, it uses the standard URI Status and Data values in *ResponseItems*.

Condition	Status	Data
Executed successfully	0 (Success)	Success
URI syntax is invalid	1 (Parse error)	Invalid URI

Condition	Status	Data
URI is not supported	6 (Internal error)	URI not found
Unable to execute URI because there currently is no active (streaming) call	6 (Internal error)	No Active Call
Unable to execute URI because the current active (streaming) call is not associated with the specified application	6 (Internal error)	No Active Call for Application
Phone is temporarily unable to execute URI due to some other transient issue	6 (Internal error)	<Failure>

Application Management URIs

These sections describe the application management URIs.

Init

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

Init 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.

Notify

The *Notify* URI generates network notifications to back-end applications. This feature is most useful for XSI objects that support action handlers (such as displayable XSI objects and RTP streams). For example, use the

Notify URI to deliver notifications to back-end applications when an XSI application is closed or when an RTP stream is terminated.

You can also specify the *Notify* URI in place of most fields that accept a generic URI, including softkeys and menu items. For example, you can call the *Notify* URI from a softkey or menu item to trigger a back-end event that does not require an interface change, such as manipulating the state of audio streams or other non-visual resources. The *Notify* URI also works in conjunction with the *QueryStringParam* URI, such that the exact contents of the *QueryStringParam* data will be used as the *Notify* URI data.

The *Notify* URI is not made in the context of an XSI application session and does not contain any HTTP cookie or session information. Thus, the back-end application cannot rely on HTTP cookies or session information to uniquely identify the client or application. Instead, the application must embed any necessary information in the *Notify* path and data fields, or leave the data field empty and rely on any default information provided by the specific event handler.



Note The *Notify* URI is not supported in the *Execute* object.

Notify URI Format

Notify:protocol:host:port:path:credentials:data

Where

protocol = network protocol to use for the *Notify* connection; http is the only supported protocol.

host = network host designated to receive the notification. Value must be entered as a hostname or IP address.

port = network port to use for the *Notify* connection. Value must be a number from 1-65535.

path = protocol-specific information. Value cannot contain colons or semicolons.

credentials = optional protocol-specific credentials used to authenticate to the server. For HTTP, this is a base64-encoded version of *userid:password*. Value cannot contain colors or semicolons. If the credentials parameter is not specified or if it is null, no Authorization header will be included in the request. The HTTP notification service will retry the request 3 times before failing and logging an error message.

data = optional application-specific event data. Value cannot contain semicolons.

Notify URI Examples

- Called from RTP *onStreamStopped* Event Handler, no credentials, with data:

```
Notify:http:myserver:8080:path/streamhandler?event=stopped:
:myStreamStoppedData
HTTP POST /path/streamhandler?event=stopped HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset="UTF-8"
Host: myserver:8080
Content-Length: 23
DATA=myStreamStoppedData
```

- Called from RTP *onStreamStopped* Event Handler, no credentials, no data:

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

```
Content-Length: 40
DATA=<notifyStreamStopped id="stream1"/>
```

- Called from *SoftKey*, with credentials, with data:

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

- Called from *SoftKey*, no credentials, no data

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

- Called from *SoftKey* with *QueryStringParam* URI:

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

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

```
HTTP POST /path/play HTTP/1.1
Accept: */*
Content-Type: application/x-www-form-urlencoded; charset="UTF-8"
Host: vmalSrvr:8080
Content-Length: 9
DATA=id=2
```

Application

The *Application* URI is a component of the Application Management API, which provides an improved hand-off between call mode and application mode. The *Application* URI allows applications to request changes to their application or window state. Applications can request to change focus, to be minimized, or to be closed.



Note The other component of the Application Management API is the Application Management Event Handler.

When an *Application* URI request is made, it has a specific application associated with it (not just the application context) and that action can only be taken on that specific application. The application specified in the *appId* parameter (of the displayable XML object) must be active at the time the action is requested, or an error will be returned.

This prevents open, but not active, applications which are buried on the application “stack” from closing the entire application context which would also close the active application, potentially disrupting the user’s interaction with the application. This also means that if an application closes or becomes non-active (for example, if user navigates out of an application, or a new application is pushed to the context) any pending *Application* URI requests are immediately cancelled.

Related Topics

[Application Event Handlers](#)

App URI Format

App:action:priority:idleTimer:applicationId

Where

action = action to be taken with the application. Values include:

- *RequestFocus*: Makes a request to the application manager to bring the application context (window) containing this application into focus (maximize). This is a request, not a demand, as higher priority applications may prevent the application from actually gaining focus. Applications must use *onAppFocusGained* event handlers to know when focus is actually gained.
 - If the requested application is Open, but not currently Active, this request will not succeed (error response).
 - If the application already has focus, the request has no effect.
- *ReleaseFocus*: Makes a request to the application manager to relinquish focus to another application context (essentially, a “move-to-back” request). Applications must use *onAppFocusLost* event handlers to know when focus is actually lost.
 - If the application does not have focus, the request has no effect.
 - If there are no other applications open (available to receive focus) then this application will retain focus.
- *Minimize*: Makes a request to the application manager to minimize the application context containing this application. This request always results in the application (eventually) being minimized. If the application has focus when this URI executes, the *onAppFocusLost* event handler will be invoked first, then the *onAppMinimize* handler.
 - If the requested application is Open, but not currently Active, this request will not succeed (error response).
 - If the application is already minimized, the request has no effect.
- *Close*: Makes a request to the application manager to close the application context containing this application.
 - If the requested application is open, but not currently active, this request will not succeed (error response). This request will result in the application context (and all applications within that context) being closed.

- If the application has focus when this URI executes, the *onAppFocusLost* event handler will be invoked prior to the *onAppClosed* event handler (which will always be invoked).

priority = priority at which the action should be take. Values include:

- 0: Do immediately, even if user is interacting with the phone. This priority is unavailable if the *Application* URI is contained within an Application Management Event Handler.
- 1: Do when user is done interacting with the phone.
- 2: Do only if the user is not interacting with the phone.

idleTimer = duration of time (in seconds) the phone or application must be idle before the action should be taken. Values must range from 10-86400 (seconds); default is 60 seconds. The *idleTimer* value has no effect on *priority=0* requests. Any pending timers are automatically cancelled when the displayable object changes for an application context.

applicationId = optional identifier of the application on which the action should be taken. Values must range in length from 1-64 string characters and cannot contain colons. The default value is the application of the displayable object in which the URI is defined.



Note If the *Application* URI is used in an *ExecuteItem*, you must specify the *applicationId* because the application context of the request cannot be inferred.

App URI Error and Response

All *Application* URI requests are asynchronous, so the only return value indicates that the URI was successfully parsed and that the specified application was valid and currently active in its context. The application is notified of the actual state change asynchronously using the event handlers.

Condition	Status	Data
Executed successfully	0 (Success)	Success
URI syntax is invalid	1 (Parse error)	Invalid URI
Unknown application ID	6 (Internal error)	Unknown Application ID
Request made to change state of an application that is not current active	6 (Internal error)	Application is not Active