



Appendix

- [Appendix A: Additional information, on page 1](#)
- [Appendix C: Products Mentioned in this Document, on page 5](#)
- [Appendix D: Terms, on page 6](#)

Appendix A: Additional information

Cisco Wireless Phone Web API and Cisco SIP Application Dependencies

The Cisco Wireless Phone Web API can send state and event information that is related to the Cisco SIP telephony Application, including:

- Call State.
- Call State Change.
- Incoming Call.
- Outgoing Call.
- On Hook.
- Off Hook.
- Line Registration Complete.
- Line Unregistration Complete.
- Phone Extension Number (DN).

Visual Design Specifications

You will likely want to make changes in your web pages due to the following:

- Different aspect ratios for the different models
- All Cisco Wireless Phone models have full touchscreens Thus your application flow may be better architected using on-page links and HTML buttons instead of pulling forward the Softkey button paradigm from the 84-Series phones.

- Screen area in pixels:
 - Cisco Wireless Phone 860 Series wireless phone is 1776x1080
 - Cisco Wireless Phone 840 Series wireless phone is 728x480

Determining the Phone Model

There are a few methods that you can use to determine the phone model:

Event Notifications

If you receive any kind of Event Notification from a Cisco Wireless Phone you can look at the outermost tag in the XML and determine if it is using the Cisco Wireless Phone Web API.

Polling the Phone

You can differentiate phone models by the header. Cisco Wireless Phones start with `<SpectralinkIPPhone>` tags.

After you receive any kind of Event Notification from a Cisco Wireless Phone, you can poll the phone for its device information that will include the Model Number of the phone. The Device Phone State Poll returns the phone model number in the `<ModelNumber>` field. (See [Receiving Device Information](#) for more details.)

User Agent

You can use the User Agent to detect which type of phone your application is talking to. Use the following values for Cisco models.

Cisco Wireless Phone

- Cisco Wireless Phone 860 == Wi-Fi no barcode
- Cisco Wireless Phone 860S == Wi-Fi + barcode
- Cisco Wireless Phone 840 == Wi-Fi no barcode
- Cisco Wireless Phone 840S == Wi-Fi + barcode

The Cisco Wireless Phone browser reports something similar to the following:

```
Mozilla/5.0 (Linux; Android 8.1.0; Cisco Wireless Phone 860 Build/OPM1.171019.026;
wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/67.0.3396.87
Mobile Safari/537.36
```

Web API Syntax Changes

The Cisco Wireless Phone browser reports something similar to the following:

```
Mozilla/5.0 (Linux; Android 8.1.0; Cisco Wireless Phone 860 Build/OPM1.171019.026;
wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/67.0.3396.87
Mobile Safari/537.36
```

- Use quotes for both the priority and the volume value. For example, `<URL priority='Normal' volume='100'>/`

Barcode Changes

The barcode interface is no longer developed as part of the Web API. See the API Spec for Cisco Wireless Phone for exact information.

Interrupt Criteria

On both wireless phones, ring and media volume are set by the user and these preferences become the default for that phone. While the phone is in DND mode, ring and media volume are affected by alerts. Neither model differentiates in the type of alert (critical, low, and so on). All types of alerts interrupt the same.

In Cisco Wireless Phone, DND has three rather self-explanatory modes: Total silence, Alarms only and Priority only. The modes for each model are altered by an alert in different ways as shown in the following table.

Mode	Model	PIVOT	Cisco Wireless Phone
Total silence	After Push XML After Push closed	Na	<ul style="list-style-type: none"> • Phone remains in Total silence. • Only visual alert. • No sound for alert and media file. • Phone remains in Total silence.
None	After Push XML After Push closed	<ul style="list-style-type: none"> • Reverts to default. • Visual alert. • Sound for alert and media file. • Remains at default. 	Na
Alarms only	After Push XML After Push closed	Na	<ul style="list-style-type: none"> • Reverts to default. • Visual alert. • Sound for alert and media file. • Remains at default.
Priority/ Priority only	After Push XML After Push closed	<ul style="list-style-type: none"> • No mode change. • Visual alert. • Sound for alert and media file. • Phone remains in Priority only. 	<ul style="list-style-type: none"> • Reverts to default. • Visual alert. • Sound for alert and media file. • Remains at default.

User Agent Change

The user agent uses the model number of the phone. This number is different for each model family. See [Determining the Phone Model](#)

Control of Soft Keys

This section provides JavaScript examples that work in conjunction with the Cisco App URLs and Cisco Alertview on Cisco Wireless Phones. These buttons and code were used in the 84-series. Your application flow may be better architected using on-page links and HTML buttons instead of pulling forward the Softkey button paradigm from the 84-series phones.

Examples below as shown in Cisco Wireless Phone.

Figure 1: The Apps URLs widget box

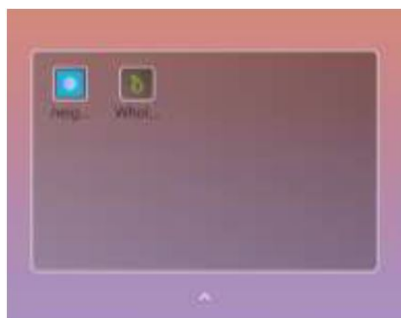
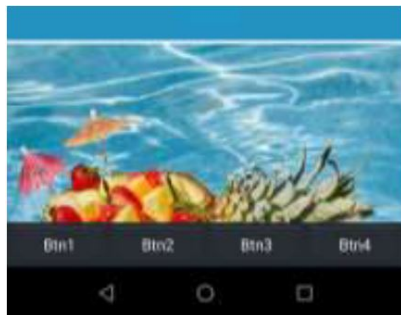


Figure 2: Softkey buttons in the browser view



The following example shows how to control soft keys to allow backwards compatibility with Cisco 84-series handsets.

Example: Soft Key Control Example for Cisco 84-Series Handsets

```
html>
<head>
<Title>Softkey JavaScript Test</Title>
<script type="text/javascript">
// PolySoftKey is the exported DOM object
// Registers a JavaScript function to be executed when a custom
softkey event occurs
PolySoftKey.connect("skCallBack");
PolySoftKey.setSoftkeyLabel(0, "One");
PolySoftKey.setSoftkeyLabel(1, "Two");
PolySoftKey.setSoftkeyLabel(2, "Three");
PolySoftKey.setSoftkeyLabel(3, "Four");
function skCallBack(key, skEvent){
```

```
switch(key) {
case 0:
document.getElementById("eventStuff").innerHTML = "SK 1 was
pressed";
break;
case 1:
document.getElementById("eventStuff").innerHTML = "SK 2 was
pressed";
break;
case 2:
document.getElementById("eventStuff").innerHTML = "SK 3 was
pressed";
break;
case 3:
document.getElementById("eventStuff").innerHTML = "SK 4 was
pressed";
break;
}
document.getElementById("eventValue").innerHTML = skEvent;
}
// hide the tool bar
function hideSKs(){
PolySoftKey.hideToolBar();
}
// show the tool bar
function showSKs(){
PolySoftKey.showToolBar();
}
</script>
</head>
<body onload="onInit()">
<div id="showButton">
<input type='button' onclick='showSKs()' value='Show Softkeys' />
</div>
<div id="hideButton">
<input type='button' onclick='hideSKs()' value='Hide Softkeys' />
</div>
<div id="eventText">
<p>Last Click: <b id='eventStuff'>0</b> </p>
<p>Event Value: <b id='eventValue'>0</b> </p>
</div>
</body>
</html>
```

Appendix C: Products Mentioned in this Document

Android is a registered trademark owned by Google, Inc.

Apache and Tomcat are trademarks of the Apache Software Foundation.

Balsamiq is a registered trademark of Balsamiq Studios, LLC.

Chrome browser is a trademark owned by Google, Inc.

Django is a registered trademark of the Django Software Foundation.

Eclipse is a trademark of Eclipse Foundation, Inc.

Firefox is a registered trademark of the Mozilla Foundation.

Fonality and trixbox are registered trademarks of NetFortris, Inc.

JavaScript is a registered trademark owned by Oracle Corporation.

PowerPoint, Visual Studio and Visio are registered trademarks of Microsoft Corporation.

Python is a registered trademark of Python Software Foundation.

Safari is a registered trademark owned by Apple Inc.

W3C, World Wide Web Consortium is a registered trademark of the Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, or Keio UniWebex phone on behalf of the World Wide Web Consortium.

Appendix D: Terms

1.0 XML API

Cisco Wireless Phone Web API

Activities

activity

Alerts

Android Notification

Android Notification message

Android Widget

custom embedded tone

Document Object Model (DOM)

Event notifications

HTTP Digest Challenge

Internal Uniform Resource Identifiers (URIs)

JavaScript

Notification bar

Phone state polling

post dialing (postd)

Push requests

Server root URL

Cisco Alertview

Cisco Configuration Management Server

Cisco App URLs

Status Bar

web application shortcut widget

App URLs

XHTML

