![Polycom logo]

Polycom® OBiPhoneXML Applications

Contents

Before You Begin

This guide describes how to develop and deploy XML-based applications to configure Polycom® VVX® Business IP Phones, OBi Edition.

The information applies to the following OBi Edition phones:

- VVX 150, OBi Edition
- VVX 250, OBi Edition
- VVX 350, OBi Edition
- VVX 450, OBi Edition

Audience, Purpose, and Required Skills

This guide is written for a technical audience involved in developing and deploying XML-based applications for OBi Edition phones. This audience includes Internet Telephony Service Providers (ITSP), Managed Service Value Added Resellers (VAR), Internet Telephony Professionals, and Technology Hobbyists.

You must be familiar with the following concepts before beginning:

- Current telecommunications practices, protocols, and principles
- Telecommunication basics, video teleconferencing, and voice or data equipment
- Open SIP networks and VoIP endpoint environments

Note for Australian readers: Throughout this document, we refer to ITSP – treat this term the same as you would for VSP (Voice Service Provider).

Note to End Users

This guide targets phone administrators who manage large installed bases.

End users are highly encouraged to use the OBiTALK web portal at www.obitalk.com to configure and manage their OBi devices, or to perform management tasks locally by using the phone's native web portal, accessible by logging into the phone from any PC with a web browser. See the *Polycom® VVX® Business IP Phones, OBi Edition User Guide* for more information about your phone's native web portal.

Getting Help

For more information about installing, configuring, and administering Polycom products, see **Documents & Software** at Polycom Support.

Polycom and Partner Resources

Polycom has a number of options available to customers who need help with their Polycom OBi Edition products. In addition to this guide, the following documents and other resources provide more details:

- Visit the OBi Edition documents and guides found at https://documents.polycom.com
  - ➢ *Polycom® VVX® Business IP Phones, OBi Edition Administration Guide*
  - ➢ *Polycom® OBi Edition Provisioning Guide*
  - ➢ *Polycom VVX Business IP Phones, OBi Edition User Guide*
- For Polycom UC Software releases and documentation, see Polycom Voice Support.
- For user guides for Polycom voice products, refer to the product support page for your phone at Polycom Voice Support.
- For help or technical support for your phones, you can search for Polycom documentation at the Polycom Unified Communications (UC) Software Resource Center.
- You can find Request for Comments (RFC) documents by entering the RFC number at http://www.ietf.org/rfc.html.
- Email the Polycom OBi Edition Service Provider Support Team at: Obi.SPSupport@Polycom.com

To find all Polycom partner solutions, see Strategic Global Partner Solutions.

The Polycom Community

The Polycom Community gives you access to the latest developer and support information. Participate in discussion forums to share ideas and solve problems with your colleagues. To register with the Polycom Community, simply create a Polycom Online account. When logged in, you can access Polycom support personnel and participate in developer and support forums to find the latest information on hardware, software, and partner solutions topics.

Documentation Feedback

We welcome your feedback to improve the quality of Polycom documentation.

You can email Documentation Feedback for any important queries or suggestions related to this documentation.

Notational Conventions

This guide provides device configuration parameters and their values in the following formats:

- Canonical fashion
- Literal fashion

Both notational conventions point to the same parameters, but their appearances are different.

The canonical fashion simplifies locating parameters on the phone's native web portal or on OBiTALK.com.

The literal fashion is required when provisioning or writing OBIPhoneXML apps.

Canonical Fashion

This example shows the format of the canonical fashion.

- ***Parameter Group Name***::**ParameterName** = `Parameter Value {replace-with-actual-value}`

The ***Parameter Group Name*** is the heading of the parameter group on the left side panel of the device local configuration or OBiTALK Configuration webpage. This string may contain spaces. When a group heading has more than one level, each level is separated with a –, such as:

- ***Services Providers - ITSP Profile A – SIP***:

The **ParameterName** is the name of the parameter as shown on the webpage and MUST NOT CONTAIN ANY SPACES. ***Parameter Group Name*** and **ParameterName** are separated by two colons (::),as shown in the first example above.

The `Parameter Value` is the literal value to assign to the named parameter and may contain spaces. You can omit ***Parameter Group Name*** or its top-level headings when the context is clear. For example:

- ***SP1 Service***::**AuthUserName** = `4082224312`
- ***ITSP Profile A - SIP***::**ProxyServer** = `sip.myserviceprovider.com`
- **ProxyServerPort** = `5082`

Literal Fashion

These examples show the format of the literal fashion. The literal fashion is used when provisioning or writing OBIPhoneXML apps.

- ***ParameterGroupName*.ParameterName**.`Parameter Value {replace-with-actual-value}`
- ***Parameter.Group.Name.ParameterGroupName*.ParameterName**.`Parameter Value`

The ***ParameterGroupName.*** is the name of the first parameter group in literal fashion. This string MUST NOT CONTAIN ANY SPACES, and always is terminated with a period, as shown. More than one ***ParameterGroupName.*** may be used. The ***ParameterGroupName.*** is case-sensitive.

The **ParameterName.** is the name of the parameter, and always is terminated with a period, as shown. This string MUST NOT CONTAIN ANY SPACES. The **ParameterName.** is case-sensitive.

The `Parameter Value` is the literal value to assign to the named parameter and may contain spaces. The `Parameter Value` is not case-sensitive, but it MUST EXACTLY MATCH the value when one or more choices are available.

When using the literal fashion in your XML, you need to exactly match the text string for ***ParameterGroupName*.ParameterName**.`Parameter Value`, but text formatting such as bold face is not required and will be removed when your script or app is processed.

Boolean Values

You can identify parameters that take a Boolean value on your phone's configuration webpages by a check box next to the parameter name. Throughout the document, we may loosely refer to a Boolean value as "enable/disable" or "yes/no", but the only valid Boolean parameter values to use in a phone configuration file is either `true/false` or `True/False` (case-sensitive). This is equivalent to selecting or clearing the check box on the configuration webpages.

Multiple Choice Values

You must provision parameters that take one of several valid options from a drop-down list on the device message with string values that match exactly one of those choices. Otherwise, the device uses the default choice. Matching the provisioned value against valid strings is case-sensitive and doesn't allow extra spaces.

Parameter Values

When entering a parameter value from the webpage or via provisioning, avoid adding extra white spaces before or after the parameter value. If the value is a comma-separated list of strings or contains attributes after a comma or semicolon, avoid adding extra white space before and after the delimiter.

For example: **CertainParameter** `= 1,2,3,4;a;b;c`

If a parameter value can include white spaces, such as **X_DisplayLabel**, use just a single space and no extra space before and after the value.

For example: **X_DisplayLabel** `= My New Service`

Getting Started

The OBi Edition VVX business IP phones provide a programmatic interface for developing third-party applications that you can download and execute on the phones. You can develop these applications (called OBiPhoneXML apps) using a proprietary mark-up language called OBiPhoneXML to create solutions involving deployed phones and servers.

Typical applications include:

- Access to corporate directories
- Voicemail navigation
- Call queue monitoring
- Integration with customer databases

XML App Execution Models

An XML app comprises one or more scripts written in OBiPhoneXML that you download and execute sequentially on the phone to achieve the desired behavior. The execution of an app starts with the first XML script, also known as the *landing script*. Depending on the interaction with the user and relevant events happening on the phone, the phone can invoke and execute additional scripts. The server that generates or stores the scripts is the (XML) App Server.

The phone uses two models to execute an XML app:

- The *pull model* is where the phone pulls the landing script from a preconfigured action URL triggered by some event on the phone (such as user pressing a feature key). The method supported by the phone for pulling is `HTTP/HTTPS GET`.
- The *push model* is where the App Server pushes the script to the phone for execution. The method supported by the phone for pushing is `SIP NOTIFY` with the Event: `obihai-xml` and Content-Type: `application/xml`.

Action URLs – Pull Model

The OBi Edition phones support the pull model by accepting the configuration of the app URL into the phone to associate with a softkey or a feature key, such that when the key is pressed, the phone downloads the corresponding script from the given URL, applies the script, and executes the script. This URL is called an Action URL.

Action URL Feature Key

To invoke an Action URL with a feature key, the feature must be set up with the function `Action URL`, and the URL of the landing script must be specified in the `Number` field of that feature key. The `Name` field of the

feature may also be configured with a friendly name to be displayed on screen to identify the functionality of the script. For example:

```
Function = Action URL
Number =
http://xmlapp-server.obihai.com:8080/xml/testmain.xml?user=jsmith&model=1
032
Name = Main Menu
```

Action URL Softkey

To invoke an Action URL with a softkey, include a softkey with the id `acturl` in a softkey set parameter, with the URL specified in a `url` attribute. For example, the default Home softkey set is: `redial, cfa,dnd,missed|lines`. You can replace the fourth softkey with an Action URL (all one line):

`redial,cfa,dnd,acturl;url="http://10.1.1.123/test.xml?user=abc&model=1032" ;label="Main Menu"`

Note that the value of the `url` attribute MUST be XML-escaped. For example, `&` must be specified as `&`.

SIP NOTIFY – Push Model

As it is typically not feasible for the App Server to send HTTP requests to the phone, the phone only accepts server-pushed XML via SIP NOTIFY requests sent to it over one of the enabled SP*n* service channels. The SIP NOTIFY must have the Event: `obihai-xml` with the Content-Type: `application/xml`.

The phone also accepts XML pushed via HTTP POST sent to the built-in Web Server port with the following URL: `http://{device-ip-address}/obihai-xml` with the Content-Type: `application/xml`.

When the XML pushed to the phone contains one or more `<ExecuteItem>` elements, the phone replies to the request with the inclusion of a message body that is an `<ObihaiIPPhoneExecuteResponse>` containing one or more `<ExecuteItem>` elements in the same order as the corresponding `<ExecuteItem>` elements in the original request. The `<ExecuteItem>` element in the response contains a result attribute that indicates the status of executing the item. It may contain additional information as a result of carrying out the `<ExecuteItem>` element.

For example, when the following XML is pushed to the phone to make a call:

```
<ObihaiIPPhoneExecute>
  <ExecuteItem id="1" URI="Dial:14089991234"/>
</ObihaiIPPhoneExecute>
```

The following message body may be included with the response returned by the phone:

```
<ObihaiIPPhoneExecuteResponse>
  <ExecuteItem id="1" result="200 OK" call-id="f234ab97"/>
</ObihaiIPPhoneExecuteResponse>
```

Event-Driven Model

The phone is programmed to perform specific actions in response to a prescribed set of events.

Events consist of:

- User actions (such as a key press)
- Telephony events (such as an incoming call)
- Server-to-phone push messages (via SIP NOTIFY Events)
- Programmed timer expirations

Actions consist of:

- Activating internal phone features
- Playing audio files and tones
- Phone screen updates
- Phone-to-server pull requests (via the HTTP protocol)

The phone configuration profile specifies the initial state and event triggering hooks for each application (such as through softkey configuration).

Typical Application Event Flow

In a typical application, a feature key is configured with an application-specific URI. When you press that key, the phone activates the URI, sending an HTTP GET request to an application server.

The server response to this request consists of an XML page, whose contents specify both visual content for the phone to display, and also actions to perform under various triggers. In this way, the phone display acts as a kind of browser on behalf of the application. User actions or other events then trigger the phone to perform a prescribed action, and possibly generate additional server requests for updated XML content.

Thus, the phone-side of the application design revolves around the XML content of messages (pages) sent by the server to the phone. These are either synchronous responses to phone HTTP requests, or else messages pushed by the server to the phone asynchronously (via SIP NOTIFY Events).

Configuring Keys to Launch Phone Apps

You can preconfigure the phone to map line keys and programmable keys to phone applications. This enables a phone user to run an application by pressing the specified key. For this use case, the key is configured with the URL from which to fetch the topmost page of the application.

Map a Line Key to a Phone App

Configure the following parameters to set `LineKey{n}` as the trigger for running an application:

Configuration Parameters for a Line Key

Configuration Parameter	Programmed Value
VoiceService.1.Phone.LineKey.{n}.Function	"Action URL"

Configuration Parameters for a Line Key

Configuration Parameter	Programmed Value
VoiceService.1.Phone.LineKey.{n}.Name	Application display name
VoiceService.1.Phone.LineKey.{n}.Number	URL from which to fetch the application's topmost XML page. The phone generates an HTTP GET request to the server when the key is pressed.

Map a Programmable Key to a Phone App

Alternatively, if a programmable key is desired as the trigger, the corresponding "`ProgKey`" entries would be configured in place of "`LineKey`".

If a sidecar key is desired as the trigger, replace "`LineKey.{n}`" with "`SideCar.{j}.Key.{k}`".

Upon receiving the XML page in response to the HTTP GET, the phone interprets the contents as described in the following sections.

Map a Softkey to a Phone App

You can also configure a softkey to invoke an application, with a similar syntax as in the following example, inserted in a softkey set parameter (under the Soft Keys configuration page):

> `acturl;url=`http://192.168.15.225:8080/obixml/testmain.xml`;label="Local Tests"`

Map a Softkey to the Line Key App

The Line Key app allows you to display all of the line keys on a single display. You can invoke this app by pressing the assigned softkey or line key.

To use a softkey, add a softkey to the Home softkey set using the following syntax:

> `acturl;url=phone://linekeysapp;label="Line Keys"`

To use a line key, select a line key and set Function=Action URL and Number=`phone://linekeysapp`

Phone Configuration Parameters

This section describes the following phone configuration elements:

- ObiPhoneXML Root Elements (Page Types)
- OBiPhoneXML Child Elements

XML Page Content and Associated Behavior

The phone recognizes several types of XML messages, and distinguishes among them based on their root element tag. Certain messages are designed to structure information on the phone display (XML pages), while others trigger phone actions or other state changes.

The following sections describe each message type. The common sub-elements (`Title`, `SoftKey`, and `IconList`), all element attributes, their values, and behavior are described in more detail in the Action URI Values section later in this document.

All OBiPhoneXML tags and attributes are case-insensitive. Unless stated otherwise, the attribute values and element values are also case-insensitive.

Compact Profile Format

OBiPhoneXML supports an alternative profile format that is more compact to reduce the file size of the profile. The element and attribute names in the full format have a corresponding short form as listed below:

- `<O>` = `<Object>`
- `<N>` = `<Name>`
- `<V>` = `<Value>`
- `<P>` = `<ParameterValueStruct>`
- `<X_R>` = `X_Reset`
- `<X_UD>` = `X_UseDefault`
- `<X_UA>` = `X_UserAccess`
- "Y"= "Yes"
- "N" = "No"

You can mix compact format and full format syntaxes in the same profile.

Common Root Element Attributes

The following tables show where the root element common attributes are used.

Common Root Element Attributes, Table 1

Attributes/ XML Root	Obihai IPPhone TextMenu	Obihai IPPhone TextScreen	Obihai IPPhone InputScreen	Obihai IPPhone Directory	Obihai IPPhone ImageFile
beep	x	x	x	x	x
timeout	x	x	x	x	x
cancelAction	x	x	x	x	x
timeoutAction	x	x	x	x	x
defaultIndex	x		x	x	
next	x	x	x	x	x
previous	x	x	x	x	x
screen-saver	x	x	x	x	x
auto-home	x	x	x	x	x
color	x	x	x	x	x
size	x	x	x	x	x
align	x	x	x	x	x
bold	x	x	x	x	x
item-height	x		x	x	
Item-bgcolor	x		x	x	

Common Root Element Attributes, Table 2

Attributes/ XML Root	Obihai IPPhone IconFileMenu	Obihai IPPhone Execute	Obihai IPPhone Status	Obihai IPPhone Configuration
beep	x	x	x	x
timeout	x			
cancelAction	x			
timeoutAction	x			
defaultIndex	x			
next	x			
previous	x			
screen-saver	x			
auto-home	x			

Common Root Element Attributes, Table 2

Attributes/ XML Root	Obihai IPPhone IconFileMenu	Obihai IPPhone Execute	Obihai IPPhone Status	Obihai IPPhone Configuration
color	x			
size	x			
align	x			
bold	x			
item-height	x			
Item-bgcolor				

The following table describes the root element common attributes.

Root Element Common Attributes

Root Element Common Attributes	Type	Description
beep	yes | no	Whether to play a short beep when page is loaded.
timeout	seconds	How many seconds to timeout the page (trigger a Cancel operation). Absent or 0 implies no timeout.
cancelAction	URL	URL to load on a Cancel Event.
timeoutAction	URL	URL to load on a Timeout Event (per timeout attribute).
defaultIndex	1-based index	Default highlighted menu item index.
next	URL	URL to load on a Next Event.
previous	URL	URL to load on a Previous Event.
screen-saver	yes | no	Set to no to stop the screen saver from firing up temporarily while the XML app is running, if screen saver is currently enabled. Default is yes.
auto-home	yes | no	Set to no to stop the phone from closing all applications and revert back to the Home screen automatically after a configured timeout without any key presses or calls. Default is yes.
color	Color	Default text color to use for all text on the page.
size	integer	Default text font size to use for all text on the page.
align	left | center | right	Default horizontal text alignment for all text on the page.
bold	yes | no	Whether to use bold font face by default for all text on the page.
item-height	integer	Default pixel height of all menu items on the page.

Root Element Common Attributes

Root Element Common Attributes	Type	Description
`item-bgcolor`	Color or Color Pattern	Default background color for all menu items on the page. Examples: • `0xffff00` • `black` • `0xffff00,0x0000ff,1` • `black, 0xffffff, 0` • `white, brown, 0`
`item-hlcolor`	Color or Color Pattern	Background color of the highlighted menu item on the page.

Common First-Level Child Elements

The following table shows the top-level elements that are common to one or more page types. The following sections show examples of these page types.

Common First-Level Child Elements

XML Root/Child	Title	Prompt	SoftKey	IconList	MenuItem
ObihaiIPPhoneTextMenu	x	x	x	x	x
ObihaiIPPhoneTextScreen	x	x	x	x	
ObihaiIPPhoneInputArea	x	x	x	x	
ObihaiIPPhoneInputScreen	x	x	x	x	
ObihaiIPPhoneDirectory	x	x	x	x	x
ObihaiIPPhoneImageFile	x	x	x		
ObihaiIPPhoneExecute					
ObihaiIPPhoneStatus					
ObihaiIPPhoneIconFileMenu	x	x	x	x	x
ObihaiIPPhoneConfiguration					

ObiPhoneXML Root Elements (Page Types)

The following sections list the page types.

- ObihaiIPPhoneTextMenu
- ObihaiIPPhoneTextScreen
- ObihaiIPPhoneInputArea
- ObihaiIPPhoneInputScreen
- ObihaiIPPhoneDirectory

- [ObihaiIPPhoneImageFile](#)
- [ObihaiIPPhoneExecute](#)
- [ObihaiIPPhoneStatus](#)
- [ObihaiIPPhoneConfiguration](#)

ObihaiIPPhoneTextMenu

The **ObihaiIPPhoneTextMenu** root element encapsulates a menu screen, consisting of a title and a scrollable list of menu entries. Each entry is paired with an action URI.

Top-level structure:

```
<ObihaiIPPhoneTextMenu>
  <Title/>
  <Prompt/>
  <MenuItem/>  <!-- one or more entries -->
  <SoftKey/>  <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneTextMenu>
```

ObihaiIPPhoneTextScreen

The **ObihaiIPPhoneTextScreen** root element encapsulates a body of free-form text.

Top-level structure:

```
<ObihaiIPPhoneTextScreen>
  <Title/>
  <Text/>
  <Prompt/>
  <SoftKey/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneTextScreen>
```

ObihaiIPPhoneInputArea

The **ObihaiIPPhoneInputArea** root element encapsulates a list of text input fields.

Top-level structure:

```
<ObihaiIPPhoneInputArea>
  <Title/>
  <Prompt/>
  <URL/> <!-- URL to submit the input results -->
  <InputArea/> <!-- exactly one entry -->
  <SoftKey/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneInputArea>
```

ObihaiIPPhoneInputScreen

The **ObihaiIPPhoneInputScreen** root element encapsulates a list of text input fields.

Top-level structure:

```
<ObihaiIPPhoneInputScreen>
  <Title/>
  <Prompt/>
  <Text/> <!-- Text to display in main window -->
  <URL/> <!-- URL to submit the input results -->
  <InputField/> <!-- one or more entries -->
  <SoftKey/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneInputScreen>
```

ObihaiIPPhoneDirectory

The **ObihaiIPPhoneDirectory** root element encapsulates a directory screen, consisting of a title and a scrollable list of directory entries. Each entry pairs a name with an action URI.

Top-level structure:

```
<ObihaiIPPhoneDirectory>
  <Title/>
  <Prompt/>
  <MenuItem/> <!-- one or more entries -->
  <SoftKey/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneDirectory>
```

ObihaiIPPhoneImageFile

The **ObihaiIPPhoneImageFile** root element encapsulates a menu screen, consisting of an image that graphically shows a set of menu options with softkeys.

Top-level structure:

```
<ObihaiIPPhoneImageFile>
  <Title/>
  <Prompt/>
  <Image/> <!-- URL to get the image file -->
  <SoftKey/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneImageFile>
```

ObihaiIPPhoneExecute

The **ObihaiIPPhoneExecute** root element encapsulates one or more action URIs for the phone to perform. This message doesn't directly change the display screen configuration, though the display may change as a result of the executed actions.

Top-level structure:

```
<ObihaiIPPhoneExecute>
  <ExecuteItem/> <!-- one or more entries -->
</ObihaiIPPhoneExecute>
```

Each `<ExecuteItem>` element takes a required URI attribute and an optional id attribute, as shown in the following example:

```
<ObihaiIPPhoneExecute beep="yes">
  <ExecuteItem URI="Dial:SP2(16073391200)" id="1"/>
</ObihaiIPPhoneExecute>
```

The URI indicates what the phone should execute. The id attribute is a reference for matching the result of corresponding `<ExecuteItem>` element in an `<ObihaiIPPhoneExecuteResponse>` returned by the phone when the `<ObihaiIPPhoneExecute>` is pushed to the phone.

ObihaiIPPhoneStatus

The **ObihaiIPPhoneStatus** root element encapsulates one or more application status informational text strings. This message doesn't change the display screen configuration.

Top-level structure:

```
<ObihaiIPPhoneStatus>
  <Session/> <!-- a text string to identify the status instance -->
  <Message/> <!-- one or more entries -->
  <IconList/>
</ObihaiIPPhoneStatus>
```

ObihaiIPPhoneConfiguration

The **ObihaiIPPhoneConfiguration** root element encapsulates one or more phone configuration parameter settings. This message doesn't change the display screen configuration.

Top-level structure:

```
<ObihaiIPPhoneConfiguration reboot="fast">
  <ConfigurationItem/> <!-- one or more entries -->
</ObihaiIPPhoneConfiguration>
```

The top-level element takes one optional attribute reboot with the following valid values:

- `"no"` – Do not reboot after processing the configuration.
- `"fast"` – Do a fast reboot after applying the configuration. This is the default reboot behavior.
- `"full"` – Do a full reboot after applying the configuration.

Each `<ConfigurationItem>` consists of:

```
<ConfigurationItem>
  <Parameter/> <!-- full parameter name -->
  <Value/> <!-- parameter value -->
</ConfigurationItem>
```

OBiPhoneXML Child Elements

These are the OBiPhoneXML child elements.

- <Title>
- <Prompt>
- <Dial> (Also Known as <Telephone>)
- <Text>
- <SoftKey>
- <IconList> and <Icon>
- <MenuItem>
- <Session>
- <Message>
- <Fn> (n = 1 – 4)

The following sections describe these elements and their usage.

<Title>

The `<Title>` element contains the title of the page. It has no attributes and no child elements. For example:

```
<Title icon="1" color="white">Corporate%0ADirectory</Title>
```

Attributes require quote marks. Special characters require escape strings, such as:

- `%0A` for newline
- `%20` for space

The format for special characters requiring escapes is a percent sign `%` followed by the two-digit ASCII hex code for the character.

<Title> Attributes

Supported Attributes	Description
`icon`	Optional attribute. Icon index corresponding to one of the icons in the `<IconList>` element of the same page.
`icon-xpos`	Optional attribute. Specifies the relative X-position of the icon's bounding box (upper left corner) within the Title area of the page.
`icon-ypos`	Optional attribute. Specifies the relative Y-position of the icon's bounding box (upper left corner) within the Title area of the page.

<Title> Attributes

Supported Attributes	Description
bgcolor	Optional attribute. Background color or color pattern for the Title area. Default is a completely transparent background color (that is, no color) such that the Title area shows the default background image underneath.
color	Optional attribute. The color to use for the text in the Title area. Default is "white".
size	Optional attribute. The font size to use for the text in the Title area. Default is "24".
bold	Optional attribute. Specifies whether to use bold font face for the text in the Title area. Value must be either "yes" or "no". Default is "no"
align	Optional attribute. Specifies the text alignment. Value must be one of "left", "center", or "right". Default is "left".
xpos	Optional attribute. Specifies the relative X-position of the text within the Title area of the page.
ypos	Optional attribute Specifies the relative Y-position of the text within the Title area of the page.

<Prompt>

The <Prompt> element may appear at the menu item level (including <MenuItem>, <InputField>, or <InputArea>, one per menu item) or at the page level (one per page).

Menu Item Level <Prompt> (One per Item)

The <Prompt> element contains the main text for the item, usually to describe the item. For example:

```
<Prompt color="white">Network%0Settings</Prompt>
```

At the Menu Item level, <Name> is an alias of <Prompt>. <Name> may be more readable when the root element is <ObihaiIPPhoneDirectory>.

Page Level <Prompt> (One per Page)

A Page Level <Prompt> displays at the bottom of the screen in a separate prompt area. For example:

```
<Prompt color="black" bold="yes">Please%0make%0a%0selection</Prompt>
```

Page Level <Prompt> Attributes

Supported Attributes	Page Level	Menu Item Level	Description
bgcolor	Y	N	Optional attribute. Background color or color pattern for the Prompt area (at the bottom of the screen).
color	Y	Y	Optional attribute. The color to use for the text in the prompt.
size	Y	Y	Optional attribute. The font size to use for the text in the prompt.

Page Level <Prompt> Attributes

Supported Attributes	Page Level	Menu Item Level	Description
bold	Y	Y	Optional attribute. Specifies whether to use bold font face for the text in the prompt. Value must be either "yes" or "no". Default is "no".
align	Y	Y	Optional attribute. Specifies the text alignment. Value must be one of "left", "center", or "right". Default is "left".
xpos	Y	Y	Optional attribute. Specifies the relative X-position of the text within the corresponding bounding area.
ypos	Y	Y	Optional attribute. Specifies the relative Y-position of the text within the corresponding bounding area.

Note: Use %XX to escape special characters: %0A for newline, %20 for space, and so forth.

<Dial> (Also Known as <Telephone>)

The <Dial> element may appear inside a <MenuItem> (also known as <DirectoryEntry>) to specify a number to dial. Note that if the containing root element is <ObihaiIPPhoneDirectory>, the <Dial> value is also shown below the <Prompt> (also known as <Name>) text on the screen for the same <MenuItem>, according to the given attributes.

For example:

```
<Telephone color="white" display="1%20(408) 999-1234">14089991234</Telephone>
```

<Dial> Attributes

Supported Attributes	Description
color	Optional attribute. The color to use for the text in the prompt.
size	Optional attribute. The font size to use for the text in the prompt.
bold	Optional attribute. Specifies whether to use bold font face for the text in the prompt. Value must be either "yes" or "no". Default is "no".
align	Optional attribute. Specifies the text alignment. Value must be one of "left", "center", or "right". Default is "left".
xpos	Optional attribute. Specifies the relative X-position of the text within the corresponding bounding area.
ypos	Optional attribute. Specifies the relative Y-position of the text within the corresponding bounding area.
display	Optional attribute. Specifies the text to display. If not specified, then the value of <Dial> displays. The value must not contain extra characters like spaces, hyphens, parentheses, and so forth.

Note: Use %XX to escape special characters: %0A for newline, %20 for space, and so forth.

<Text>

The `<Text>` element contains the text to display in the main window of an `<ObihaiIPPhoneTextScreen>` app. For example:

```
<Text color="white" bgcolor="black,0x454445,0">Hello,%0AIs%0Ait%0Ame%0Ayou're%0A
ooking%0Afor?</Text>
```

<Text> Attributes

Supported Attributes	Description
bgcolor	Optional attribute. Background color or color pattern for the Text area. Default is "white".
color	Optional attribute. The color to use for the text in the prompt.
size	Optional attribute. The font size to use for the text in the prompt.
bold	Optional attribute. Specifies whether to use bold font face for the text in the prompt. Value must be either "yes" or "no". Default is "no".
align	Optional attribute. Specifies the text alignment. Value must be one of "left", "center", or "right". Default is "left".

Note: Use `%XX` to escape special characters: `%0A` for newline, `%20` for space, and so forth.

<SoftKey>

Each `<SoftKey>` element configures one softkey. Each XML page can contain multiple `<SoftKey>` elements.

<SoftKey> Index Attribute

Supported Attribute	Description
index	Required. A 1-based index where index="1" corresponds to the leftmost softkey on screen within the current page.

<SoftKey> Child Elements

Supported Child Elements	Description
<Label>	Optional element. A `<Label>` element contains the text to display for this softkey. It has no attributes or child elements.
<URI>	Required element. The URL to execute when this softkey is pressed.

This example shows the structure of the `<SoftKey>` element.

```
<SoftKey index="1">
   <Label>Submit Form</Label>
   <URI>SoftKey:Submit</URI>
</SoftKey>
```

<IconList> and <Icon>

The `<IconList>` element is a container only that consists of one or more `<Icon>` elements:

```
<IconList>
   <Icon/>  <!-- one or more entries -->
</IconList>
```

Each `<Icon>` element contains the URL of an icon image file to be used as the icon referenced elsewhere in the current page (for example, when rendering a `<MenuItem icon="x">` element). `<Icon>` has no child elements. The icon URL can be an external URL or the name of an internal icon listed in the Built-In Icons section.

<IconList> and <Icon> Attribute

Supported Attribute	Description
`index`	Required attribute. A unique string ID (usually a numerical value) that references the icon within the current page.

This example shows the structure of these elements:

```
<IconList>
   <Icon index="1">http://abc.com/speaker.png</Icon>
   <Icon index="2">http://abc.com/telephone.gif</Icon>
   <Icon index="3">http://abc.com/house.jpg</Icon>
</IconList>
```

<InputField> and <InputArea>

The `<InputField>` and `<InputArea>` elements can be used only inside an `<ObihaiIPPhoneInputScreen>` element. Each element represents one input parameter to be collected from the user. `<InputField>` is for single-line input only. `<InputArea>` is for multiline input. Otherwise, both elements use the same syntax.

This example shows the structure for these elements:

```
<InputField icon="3" type="string" password="no" editable="yes">
   <Prompt>First Name</Prompt>
   <Parameter>firstName</Parameter>
   <Default></Default>
   <InputFlags>AU</InputFlags>
</InputField>
```

<InputField> and <InputArea> Attributes

Supported Attributes	Description
icon	Optional attribute. The index of the `<icon>` element on the same page to be used for this item. Default is no icon.
type	Optional attribute. Valid values are: • `"string"` – Normal alphanumeric string. • `"number"` – Numerical value (default). • `"IP"` – IP address.
password	Optional attribute. Valid values are: • `"yes"` – Field is a password. • `"no"` – Field is not a password (default).
editable	Optional attribute. Valid values are: • `"yes"` – Field is editable (default). • `"no"` – Field is read-only.
input-xpos	Optional attribute. Relative pixel X-position of the input box (upper left corner) within the menu-item on the InputScreen.
input-ypos	Optional attribute. Relative pixel Y-position of the input box (upper left corner) within the menu-item on the InputScreen.
input-height	Optional attribute. Specifies the pixel height of the input box within the menu-item on the InputScreen.
input-width	Optional attribute. Specifies the pixel width of the input box within the menu-item on the InputScreen.
input-color	Optional attribute. Specifies the text color (such as `"0xffff00"`) to use for the text inside the input box.
input-size	Optional attribute. Specifies the text font size (such as `"12"`) to use for the text inside the input box.
input-bold	Optional attribute. Specifies whether bold font face is used for the text inside the input box. Value must be either `"yes"` or `"no"`.
input-bgcolor	Optional attribute. Specifies the background color or color pattern to use for the input box. For example: • `"black"` • `"0,0xffffff,1"` • `"white,yellow,1"`
icon-xpos	Optional attribute. Specifies the relative X-position of the icon's bounding box (upper left corner) within the Title area of the page.
icon-ypos	Optional attribute. Specifies the relative Y-position of the icon's bounding box (upper left corner) within the Title area of the page.

<InputField> and <InputArea> Attributes

Supported Attributes	Description
bgcolor	Optional attribute. Background color or color pattern for the input menu item.
height	Optional attribute. Total pixel height of the input menu-item that encloses an optional prompt and an input box.

<InputField> and <InputArea> Child Elements

Supported Child Elements	Description
<Prompt>	Optional element. The text to display for this menu item (which includes an input box). By default, the prompt text is shown right above the input box of the same menu item.
<Parameter>	Required element. The URL query parameter corresponding to this input field.
<Default>	Optional element. Initial value for this field.
<Inputflags>	Optional element. The following flags can be used in combination (case insensitive): • A – Alphanumeric string • N – Number (the default) • T – Telephone Number • U – Uppercase Only • L – Lowercase Only • P – Password

<MenuItem>

Each element defines an item in a table to be rendered on the screen.

This example shows the structure for the <MenuItem> element:

```
<MenuItem icon="3">
  <Prompt>Do Not Disturb</Prompt>
  <URI>http://abc.com/dnd.php?dnd=1</URI>
  <Dial>*78</Dial>
  <Selection>dnd-item</Selection>
</MenuItem>
```

Note: <DirectoryEntry> is an alias of <MenuItem>. You may use either element.

\<MenuItem\> Attributes

Supported Attributes	Description
`icon`	Optional attribute. The index of the `<icon>` element on the same page used for this item.
`height`	Optional attribute. The pixel height of the menu item (overwrites the default or inherited item height).
`bgcolor`	Optional attribute. The background color or color pattern for the menu item (overwrites the default or inherited item background color).
`icon-xpos`	Optional attribute. The relative X-position of the icon's bounding box (upper left corner) within the menu item.
`icon-ypos`	Optional attribute. The relative Y-position of the icon's bounding box (upper left corner) within the menu item.

Supported Child Elements of \<MenuItem\>

Supported Child Elements	Description
`<Prompt>` or `<Name>`	The text to display for this menu item. For `<ObihaiIPPhoneDirectory>`, the text is usually the name of the directory entry.
`<URI>`	The URI to execute when this item is clicked/selected
`<Dial>` or `<Telephone>`	The number to dial when executing `SoftKey:Dial` or when the phone is off hook while the item is highlighted. When the root element is `<ObihaiIPPhoneDirectory>` the number to dial is also shown for the menu item below the `<Prompt>` text.

Supported Child Elements of <MenuItem>

Supported Child Elements	Description
<Selection>	If specified, the query string `selection={value}` is appended to the URI when executing the URI while this item is highlighted, where `{value}` is the value of this element. Note: `{value}` must be properly url-encoded and XML-escaped. The phone will use the given value as parameters in the URL

<InstallURI>	This is an optional element that, when specified, contains the URL to download a data resource and the target internal path to store the downloaded data. General syntax is `<InstallURI>{target}={url}</InstallURI>`. For example:
	
	`<InstallURI>`
	`/backgnd/CherryBlossom.jpg=http://abc.com/pictures/cherry-blossom.jpg`
	`</InstallURI>`
	
	The following are the only acceptable target folders:
	• `/usr` (for general user data)
	• `/ringtones` (for ringtones you can choose from UI)
	• `/backgnd` (for wallpaper pictures you can choose from UI)
	• `/dict` (for custom dictionary files you can choose a Language from UI)
	• `/fonts` (for custom text font files you can choose from UI)
	• `/people` (for phone book pictures you can assign to each contact)
	
	`<InstallURI>` typically is triggered by pressing the `<SoftKey>` that has `<URI>Install:Select</URI>` specified.

<Session>

This element has no attributes and no child elements. This string identifies an instance of status message, so it can be updated or cleared by the server. It can be used only inside the `<ObihaiIPPhoneStatus>` element.

Example:

```
<Session>dnd-status</Session>
```

<Message>

This element has no child elements and can be used only inside an `<ObihaiIPPhoneStatus>` element. It contains the text message to be displayed in the status area of the phone screen.

Example:

```
<Message icon="3">DND Enabled</Message>
```

Supported Child Elements of <Message>

Supported Attributes	Description
icon	The index of the `<icon>` element on the same page used for this item.
timeout	Timeout in seconds for displaying the given message. Absent or 0 timeout makes the message display until the server clears it.

<Fn> (n = 1 − 4)

Anonymous fields with no defined attributes are referred to as `$Fn` with a corresponding value of n inside a `<ScreenItem>` element. The macro expands into the corresponding value of `<Fn>` when applying the `<ScreenItem>` element on each `<MenuItem>` element. See the Customizing the Screen Layout section for usage examples of these elements.

Screen Appearance Parameters

This section describes how you can customize your phone's screen using ObiPhoneXML.

When authoring an OBiPhoneXML App, you need to consider the actual screen resolution for rendering on different phone models. By default configuration, when your phone requests an OBiPhoneXML, it includes its model number in the User-Agent header of the HTTP requests. See the VVX Phone Screen Resolution table later in this section for more information on phone screen resolutions.

Screen Background Colors

You can customize your screen backgrounds as solid colors, patterned colors, or displayed images.

Color Values

A color value is an integral value expressed in RGB or by HTML name. In RGB, the first byte is the blue value (0-255), the next byte is the green value (0-255), and the last byte is the red value (0-255). The rest of the bytes aren't used, and you must set them to 0.

When expressed in hex, the first three RGB values are concatenated. When expressed as decimal, the value is the hex value converted to decimal.

Examples of color values include 0, 3194703, 0xff0000, and 0xffff00.

You can also use a predefined color name instead of an RGB value. The ObiPhoneXML software includes the predefined color names listed in the following table.

Example Mappings Between Color Names, Hex Color Values, and RGB Values

Color Name	Hex Value	Decimal Value	RGB Value
black	0x000000	0	rgb(0, 0, 0)
white	0xffffff	16777215	rgb(255, 255, 255)
red	0xff0000	16711680	rgb(255, 0, 0)
green	0x008000	32768	rgb(0, 128, 0)
blue	0x0000ff	255	rgb(0, 0, 255)
cyan	0x00ffff	65535	rgb(0, 255, 255)
magenta	0xff00ff	16711935	rgb(255, 0, 255)
yellow	0xffff00	16776960	rgb(255, 255, 0)

Example Mappings Between Color Names, Hex Color Values, and RGB Values

Color Name	Hex Value	Decimal Value	RGB Value
brown	0xa52a2a	10824234	rgb(165, 42, 42)
lightred	0xffcccb	16764107	rgb(255, 204, 203)
lightgreen	0x90ee90	9498256	rgb(144, 238, 144)
lightblue	0xadd8e6	11393254	rgb(173, 216, 230)
lightcyan	0xe0ffff	14745599	rgb(224, 255, 255)
lightmagenta	0xe78be7	15174631	rgb(231, 139, 231)
lightgray	0xd3d3d3	13882323	rgb(211, 211, 211)
darkgray	0xa9a9a9	11119017	rgb(169, 169, 169)

Color Patterns

A color pattern is a set of three comma-separated values that specifies a starting color, an ending color, and a direction.

The phone renders the color by displaying the starting color at one bounding edge and blends the color to the ending color at the opposite bounding edge. You can use a value of 0 for the starting or ending color to display a single color. For example, if you specify only the starting color and use 0 for the ending color value, the phone displays the starting color from edge to edge.

The direction value specifies the direction of the color gradient. 0 displays the gradient from top to bottom, and 1 displays the gradient from left to right. If you don't specify the direction, the default is 0 (top to bottom).

Example Color Patterns

Starting Color	Ending Color	Direction	Color Pattern Description
0x000000 (black)	0xffffff (white)	1	Fades horizontally from black to white.
0xff0000 (red)	0	—	Solid red.
0	0x90ee90 (lightgreen)	0	Solid light green.
0x0000ff (blue)	0xffff00 (yellow)	— (defaults to 0)	Fades vertically from blue to yellow.

Background Images

A background image is specified by the <bgimg> attribute. This attribute's value can be a filename for an image stored on your phone, or it can be a valid URL that specifies where the image is stored on the internet. The image must be a JPEG, PNG, BMP, or GIF, with a recommended size of 800 x 480 pixels. PNG files provide the best scalability. For JPEGs, the maximum allowable size is 1024 x 768 pixels.

For all images, the best resolution is the native phone screen size, as shown in the following table:

VVX Phone Screen Resolution

Phone Model	Screen Resolution (WxH), Pixels	Color Description
VVX 150	132 x 64	monochrome
VVX 250	320 x 240	color
VVX 350	320 x 240	color
VVX 450	480 x 272	color

Since the VVX 150 has a monochrome screen, it correctly interprets only black (`0x000000`) and white (`0xffffff`), so these are the only colors that should be used for this phone model. See Attributes in <ScreenItem> XML for more information.

Customizing the Screen Layout

You can customize the layout of the widgets on the screen by embedding a `<ScreenItem>` element in a `CDATA` block for the following OBi Edition phone app types:

- `<ObihaiIPPhoneTextMenu>`
- `<ObihaiIPPhoneIconFileMenu>`
- `<ObihaiIPPhoneDirectory>`

The `<ScreenItem>` element has the same syntax as the one you use to configure the `CallItem` and `RingItem` parameters on the phone.

For example:

```
<ObihaiIPPhoneDirectory next="http://192.168.15.225:8080/obixml/dir-p2.xml"
                        previous="http://192.168.15.225:8080/obixml/dir-p3.xml">
  <![CDATA[
  <ScreenItem bgcolor="0x888888" hlcolor="0xffffff" height="156" height2="60">
    <setvar name="cid"
      value="http://abc-pub.com/cid-var-tree.php?name=$prompt&amp;tel=$url"/>
    <!-- A 20-pixel high caption for each item -->
    <span height="20" bgcolor="0x11cccc" textcolor="0xffffff">
      <text valign="center" align="left" size="16" font="@gfont-bold">$F1</text>
    </span>
<span ypos="20">
      <!-- If picture url (stored in F2) is available, show it -->
      <img width="40" height="40" src="$f2" valign="center"
display="$eval({$f2}!=)"/>
<!-- Otherwise, show a generic nopic icon -->
      <img width="40" height="40" src="http://xyz.com/pics/nopic.png"
        valign="center" display="$eval({$f2}==)"/>
```

```
      <!-- Show this if nitems < 2 -->
      <span xpos="44" display="$eval($nitems&lt;2)">
        <text ypos="0" size="16" font="@gfont-bold" textcolor="0x000000">Name:
$prompt</text>
        <text ypos="18" size="14" font="@gfont" textcolor="0x000000">Tel:
$url</text>
        <text ypos="36" size="14" font="@gfont" textcolor="0x000000">Org:
$cid.org</text>
      </span>

      <!--(Otherwise) Show this if nitems > 1 -->
      <span xpos="44" display="$eval($nitems&gt;1)">
        <text ypos="0" size="16" font="@gfont-bold" textcolor="0x000000">Name:
$prompt</text>
        <text ypos="18" size="14" font="@gfont" textcolor="0x000000">Tel:
$url</text>
      </span>
    </span>
  </ScreenItem>
  ]]>
  <Title>Customized Directory (P1/15)</Title>
  <MenuItem>
    <Prompt>Smiling Face</Prompt>
    <URI>103</URI>
    <F1>1/147</F1>
    <F2>http://xyz.com/pics/Smiling.Face.png</F2>
  </MenuItem>
  <MenuItem>
    <Prompt>Cynthia M. Locke</Prompt>
    <URI>102</URI>
    <F1>2/147</F1>
    <F2>http://xyz.com/pics/cindy.locke.png</F2>
  </MenuItem>
  <MenuItem>
    <Prompt>Whatever K. Xyzel</Prompt>
    <URI>101</URI>
    <F1>3/147</F1>
  </MenuItem>
  <SoftKey index="1">
    <Label>Call</Label>
    <URI exit="yes">SoftKey:Dial</URI>
  </SoftKey>
  <SoftKey index="2">
    <Label>Previous</Label>
```

```
                <URI>SoftKey:Previous</URI>
           </SoftKey>
           <SoftKey index="3">
               <Label>Next</Label>
               <URI>SoftKey:Next</URI>
           </SoftKey>
      </ObihaiIPPhoneDirectory>
```

where `http://abc-pub.com/cid-var-tree.php?name=Joe%20Smith&tel=4089991234` returns a "var-tree" XML document such as this:

```
      <cid value="1">
         <org value="ABC Publishing, Inc.">
             <title value="VP, Sales"/>
         </org>
         <name value="Joe Smith"/>
      </cid>
```

<ScreenItem> XML

The following sections describe the elements, attributes, and macros in a `<ScreenItem>` XML file.

Note: The following elements are not supported on the VVX 150 because of its lower screen resolution:

- `<img>`
- `<Icon>`
- `<IconList>`
- `<InputArea>`
- `<ScreenItem>`
- `<ObihaiIPPhoneImageFile>`
- `<ObihaiIPPhoneIconFileMenu>`
- `<ObihaiIPPhoneInputArea>`

For best text rendering on the VVX 150, use font-size 9 only (9 is the default).

The following table lists the elements in a `<ScreenItem>` XML file:

Elements in <ScreenItem> XML

Element	Attributes	Description
`<ScreenItem>`	Required attribute: `height` Optional attributes: `bgcolor`, `hlcolor`, `height2`	The root element. The highlighted screen item uses the `hlcolor` attribute as its background color. Other screen items use the `bgcolor` attribute as their background color. If `bgcolor` isn't specified, the default value is `0xffffff` (white). If `hlcolor` isn't specified, it takes the same value as `bgcolor`. The height attribute determines the height (in number of pixels) of the item on screen. The maximum value for the height attribute is device-specific, as shown: • VVX 150: 64 pixels, monochrome • VVX 250: 240 pixels, color • VVX 350: 240 pixels, color • VVX 450: 272 pixels, color
`<span>`	Required attribute: `height` Optional attributes: `width`, `xpos`, `ypos`, `bgcolor`, `textcolor`, `font`, `size`	`<span>` defines a rectangular region with its own background color (that overrides the `bgcolor`/`hlcolor` attribute of the screen item). For example, it can be used to define a caption area for the `<ScreenItem>` element.
`<img>`	Required attribute: `src` Optional attributes: `width`, `height`, `xpos`, `ypos`, `resize`	`<img>` specifies a rectangular region and a URL of a picture to display in that rectangular region.
`<text>`	Required attribute: none Optional: `width`, `height`, `xpos`, `ypos`, `size`, `font`, `textcolor`	Use ` ` to insert a white space.

Attributes in <ScreenItem> XML

In XML, the `<ScreenItem>` element uses the attributes described in the following table:

Attributes in <ScreenItem> Element XML

Attribute	Elements	Description
`height`	`<ScreenItem>`, `<span>`, `<text>`, `<img>`	Height of the item's bounding rectangle in pixels.
`height2`	`<ScreenItem>`	Height of the item's bounding rectangle in pixels, when the number of items is greater than 1. If the value is not specified, the value of the height attribute is used for any number of items.

Attributes in <ScreenItem> Element XML

Attribute	Elements	Description
width	<span>, <text>, <img>	Width of the bounding rectangle in pixels.
xpos	<span>, <text>, <img>	x-offset from the upper left corner of the parent bounding rectangle in pixels.
ypos	<span>, <text>, <img>	y-offset from the upper left corner of the parent bounding rectangle in pixels.
bgcolor	<ScreenItem>, <span>	Background color of the element in RGB value expressed as an integer with the Blue component in the lower 8 bits, the Green component in the next 8 bits, and the Red component in the top 8 bits. For example: "0xff0000" for RED, "0x00ff00" for GREEN, and "255" for BLUE. The bgcolor attribute also can be expressed in gradient color format with a starting color, ending color, and a direction (0 for vertical and 1 for horizontal). For example: "0xaaaaaa,0xdddddd,1".
hlcolor	<ScreenItem>	Background color to use when the item is highlighted. If not specified, it is taken as the same as the bgcolor attribute.
bgimg	<ScreenItem>, <span>	Background image to fill the bounding rectangle. The value can be the full path of a locally stored picture, or an http/https URL. If this value is specified, bgcolor and hlcolor are ignored.
hlimg	<ScreenItem>	Background image to fill the bounding rectangle when the <ScreenItem> element is highlighted. The value can be the full path of a locally stored picture, or an http/https URL. If the value isn't specified, the value of bgimg is used. If either hlimg or bgimg is specified, hlcolor is ignored.
textcolor	<span>, <text>	Foreground color to render the display text
font	<span>, <text>	The font to use to render the display text. The only allowed values are: @gfont and @gfont-bold.
size	<span>, <text>	The font size (in pixel height) to use, such as: 12, 14, 18.
wrap	<text>	The value should evaluate to "0" or "1" to determine if the text should auto-wrap to the next line when running out of space on the current line.
resize	<text>, <img>	The value should evaluate to "0" or "1" to determine if the text should auto-shrink to fit on the current line.

Attributes in <ScreenItem> Element XML

Attribute	Elements	Description
`align`	`<text>`, `<img>`	Specifies the horizontal alignment of the widget within the parent rectangle. Valid values are: • "`left`" • "`right`" • "`center`" Default is "`left`".
`valign`	`<text>`, `<img>`	Specifies the vertical alignment of the widget within the parent rectangle. Valid values are: • "`top`" • "`bottom`" • "`center`" Default is "`top`".
`ns`	`<dict>`	Specifies the name space to search for a dictionary entry for the given phrase.
`src`	`<img>`	The value should be either an internal picture file or an http/https URL of an image file, such as `http://abc.com/cid_pics/user-pic?number=$number`
`display`	`<span>`, `<text>`, `<img>`	Value should be evaluated to a "0" to hide the element, or "1" to show the element on the screen.

Macros That Can be Used Inside a <ScreenItem> Element

The following macros can be used inside a `<ScreenItem>` element:

<ScreenItem> Macros

Macro	Where Used	Description
`$nitems`	Anywhere inside <ScreenItem>.	Number of items currently shown on the screen.
`$name`	Anywhere inside <ScreenItem>. Usually used inside a `<text>` element or `src` attribute.	For a CallItem or RingItem, this expands into the peer's Caller-ID, or an empty string if Caller-ID Name is not available. For other types of `<ScreenItem>`, this expands into an empty string.
`$number`	Anywhere inside <ScreenItem>. Usually used inside a `<text>` element or `src` attribute.	For a CallItem or RingItem, this expands into the peer's Caller-ID Number, or an empty string if the Caller-ID Number is not available. For other types of `<ScreenItem>`, this expands into an empty string.

<ScreenItem> Macros

Macro	Where Used	Description
$pic	Anywhere inside <ScreenItem>. Usually used inside a src attribute.	For a CallItem or RingItem, this expands into the path/URL to get the Caller-ID Picture, or an empty string if Caller-ID Picture is not available. Calller-ID picture information is extracted from SIP signaling messages. For other types of <ScreenItem>, this expands into an empty string.
$org	Anywhere inside <ScreenItem>. Usually inside a <text> element.	For a CallItem or RingItem, this expands into the peer's organization information that is extracted from SIP signaling messages, or an empty string if organization information is not available. For other types of <ScreeItem>, this expands into an empty string.
$timer	Anywhere inside <ScreenItem>. Usually inside a <text> element.	For a CallItem or RingItem, this expands into the call timer string that counts the duration of the call, displaying the call duration in hh:mm:ss format on the screen, at 1 second resolution. For other types of <ScreenItem>, this expands into an empty string.
$state	Anywhere inside <ScreenItem>. Usually inside a <text> or <dict> element.	For a CallItem or RingItem, this expands into the current call state, which can be one of the following values: • Trying • Peer Ringing • Ringing • Connected • Holding • Call Ended • Incoming Page • Line Seize • SCA In Use • Call Parked For other types of <ScreenItem>, this expands into an empty string.
$stateIcon	Anywhere inside <ScreenItem>. Usually inside a src attribute.	For a CallItem or RingItem, this expands into the internal icon file that represents the current call state. For other types of <ScreenItem>, this expands into an empty string.

<ScreenItem> Macros

Macro	Where Used	Description
`$index`	Anywhere inside `<ScreenItem>`. Usually inside a `<text>` element.	For a CallItem or RingItem, this expands into the index information of the item, such as `1/2`, `2/2`, and so forth. This is only applicable if the call item is grouped under a Line Key that controls multiple call appearances. If the underlying Line Key controls just one call appearance, it is an empty string.
`$coachIcon`	Anywhere inside `<ScreenItem>`. Usually inside a `src` attribute.	For a CallItem or RingItem, this expands into an internal icon that indicates if the current call is in Coaching mode or Coachee mode. The icon is hidden if the call is in neither mode.
`$eval`	Inside any attributes of any element inside `<ScreenItem>`. Often used inside a display attribute to conditional show/hide a block of elements on the screen.	Evaluates a simple or complex boolean expression to produce either "0" for false or "1" for true. For example: • `$eval({$pic}==)` evaluates to "1" if `$pic` after expansion is an empty string, or "0" otherwise. • `$eval($nitems<2)` evaluates to "1" if `$nitems` after expansion is less than 2 numerically, or "0" otherwise. A simple expression is one that is just one operator with two operands, such as `$a==3`. You can enclose a simple operand with optional parentheses, such that `$eval($a==3)` and `$eval(($a==3))` are equivalent. A complex expression comprises more than one simple expression. Each simple expression MUST be enclosed in parentheses in a complex expression, such as: `$eval(($a==3)&&($b==yes)).` No extra white spaces are allowed anywhere in the expression. For macros that potentially can can expand into an empty string or a string that contains characters reserved for operations (=, !, >, <, +, -, *, and /), you must enclose the entire macro with braces { }. Otherwise, the braces are optional. You normally use `$eval()` in a display attribute to show just one of several overlapping elements where the expressions would evaluate to "1" for one and only one of the elements. For example: `$eval({$pic}==)` and `$eval({$pic!}=)` are mutually exclusive. Same for `$eval($nitems<2)` and `$eval($nitems>1)`

<ScreenItem> Macros

Macro	Where Used	Description
`$eval` (continued)	Inside any attributes of any element inside `<ScreenItem>`. Often used inside a display attribute to conditional show/hide a block of elements on the screen.	The following operands for `$eval` are supported: • `==` equal (character for character) • `!=` not equal (character for character) • `>` greater than (numerically) • `<` less than (numerically) • `>=` greater than or equal to (numerically) • `<=` less than or equal to (numerically) • `*` integer multiply • `+` nteger add • `−` integer substract • `&&` logical AND • `||` logical OR Note that `>`, `<`, and `&` must be properly escaped in an XML document.
`$prompt`	Anywhere inside `<ScreenItem>`. Usually used inside a `<text>` element or `src` attribute.	For a MenuItem in an OBiPhone XML app, this expands into the value of the `<Prompt>` element inside the MenuItem.
`$url`	Anywhere inside `<ScreenItem>`. Usually inside a `<text>` element or `src` attribute.	For a MenuItem in an OBiPhone XML app, this expands into the value of the `<Dial>` element if present, or the `<URL>` or `<URI>` element inside the MenuItem.
`$icon`	Anywhere inside `<ScreenItem>`. Usually inside a `src` attribute.	For a MenuItem in an OBiPhone XML app, this expands into the icon (in an IconList) referenced in the icon attribute of the MenuItem.
`$f`*n* *n* = 1 − 4	Anywhere inside `<ScreenItem>`. Usually inside a `<text>` element or `src` attribute.	For a MenuItem in an OBiPhone XML app, this expands into the value of the corresponding `<F`*n*`>` element inside the MenuItem.

Cache Control of Downloaded (Temporary) Data

The phone can download and cache the following data items at run time:

- Ring tone file from the URL stored in **User Preferences>DefaultRingtone**
- Picture file from the URL stored in **User Preferences>BackgroundPicture**
- Caller ID Picture files from URL extracted from Call-Info header of SIP Messages
- Picture files from URL in `src` attribute of `<img>` elements in rendered `<ScreenItem>`
- Var-Tree XML files from URL in `value` attribute of `<setvar>` elements in rendered `<ScreenItem>`
- Icon, .wav, or ,mp3 files when rendering OBiPhone XML apps

The phone gets the expiration time of each individually cached data file from the HTTP/Cache-Control header in the 200 response when the phone originally downloads the file from the server. If the Cache-Control header exists in the 200 response with the max-age attribute specified, the phone caches the downloaded data file for the number of seconds as specified in the max-age attribute. Otherwise the

phone caches the data file indefinitely or the phone clears all the cached data in one of the following scenarios:

- Phone power cycles
- After a full reboot (such as after a firmware update)
- User selects **Clear Data Cache** from the **Settings App** from the phone UI
- User enables the **Phone Settings>ClearDownloadedDataCache** option on the webpage and submits the change
- Administrator provisions the phone with the value of the **VoiceService.1.Phone.ClearDownloadedDataCache** parameter set to `true`

Action URI Values

Action URI values specify the actions that the phone performs. The phone accepts these values as the enclosed values of `<URI>` XML elements and also as the values of attributes that expect an action URI, such as `cancelAction`. This section describes each action URI and the associated behavior.

Note that some `<URI>` values take an optional exit attribute that can be set to "`yes`" or "`no`" (default is "`no`"). If exit is present and equal to "`yes`", the current XML app exits right before executing the current `<URI>`. The following table also indicates which `<URI>` allows the exit attribute.

Action URI Values

Action URI	Description	Allow Exit Attribute
`Dial:{number}[;name={callee-name}][;pic={callee-pic-url}]`	Dials the given {number}. In addition, you can specify two optional parameters. The name parameter represents the callee name that the phone should display on the screen. The pic parameter represents the URL of a picture file (png, jpeg, gif, or bmp) that the phone can show on the screen as the callee's picture. You can enclose `{callee-name}` and `{callee-pic-url}` inside a pair of double quotes if necessary.	Y
`Key:Keypad{x}`	Generates a key press event for the key `{x}` where `{x}` is a keypad digit: 0, 1, 2, 3, ..., 9.	Y
`Key:KeypadPound`	Generates a key press event for the pound (#) key `{x}`.	Y
`Key:KeypadStar`	Generates a key press event for the star (*) key.	Y

Action URI Values

Action URI	Description	Allow Exit Attribute
`Key:{key}`	`{key}` is one of the following hard keys: Soft1 Soft Key 1 Soft2 Soft Key 2 Soft3 Soft Key 3 Soft4 Soft Key 4 NavUp Navigation Key Up NavDwn Navigation Key Down NavLeft Navigation Key Left NavRight Navigation Key Right Goodbye Home Key Select Select (OK) Key Cancel Cancel (Back) Key KeyPad0 Keypad 0 KeyPad1 Keypad 1 KeyPad2 Keypad 2 KeyPad3 Keypad 3 KeyPad4 Keypad 4 KeyPad5 Keypad 5 KeyPad6 Keypad 6 KeyPad7 Keypad 7 KeyPad8 Keypad 8 KeyPad9 Keypad 9 KeyPadStar Keypad Star (*) KeyPadPound Keypad Pound (#) Mute MUTE Key VolUp VOLUME UP Key VolDwn VOLUME DOWN Key Speaker SPEAKER Key Headset HEADSET Key	Y

Action URI Values

Action URI	Description	Allow Exit Attribute
`Key:{key}`	`{key}` is one of the following hard keys: • SoftKey1 LINE Key 1 • SoftKey2 LINE Key 2 • SoftKey3 LINE Key 3 • SoftKey4 LINE Key 4 • SoftKey5 LINE Key 5 • SoftKey6 LINE Key 6 • SoftKey7 LINE Key 7 • SoftKey8 LINE Key 8 • SoftKey9 LINE Key 9 • SoftKey10 LINE Key 10 • SoftKey11 LINE Key 11 • SoftKey12 LINE Key 12 Note: Number of line keys is phone model-specific.	Y

`Led:Softkey{n}={led-pattern}`
`{n} = 1,2,3,4`
`Led:LineKey{n}={led-pattern}`
`{n} = 1,2,..12`
	Sets the LED pattern for Line Key `{n}`, where `{led-pattern}` =
	• `off` – Off
	• `on` – Steady red
	• `red` – Steady red
	• `green` – Steady green
	• `slowflash` – Slow flashing in red
	• `red-slowflash` – Slow flashing in red
	• `green-slowflash` – Slow flashing in green
	• `fastflash` – Fast flashing in red
	• `red-fastflash` – Fast flashing in red
		Y

`Led:ExpMod{m}SoftKey{n}={led-pattern}`
`{m} = 1, 2`
`{n} = 1, 2, .., 16`
	Sets the LED pattern for key `{n}` in Side Car `{m}`.	Y

`SoftKey:BackSpace`	Enters a backspace in the current input field.	
`SoftKey:ChangeMode`	Toggles dialpad entry mode between digit and letters.	
`SoftKey:Dial` `SoftKey:Dial2`	Dials the number that is the contents of the `Selection` or `Dial` element of the highlighted item.	Y
`SoftKey:Dot`	Enters a dot (`.`) character in the current input field.	
`SoftKey:Exit`	Exits the app.	
`SoftKey:Next`	Executes the URL associated with the `Next` event.	
`SoftKey:NextSpace`	Enters a space character in the current input field.	

Action URI Values

Action URI	Description	Allow Exit Attribute
`SoftKey:Previous`	Executes the URL associated with the Previous event.	
`SoftKey:Select`	Clicks the highlighted item.	
`SoftKey:Submit`	Executes the URL of the current input screen with the parameter=value pairs of all the parameters appended to the URL.	
`Wav.Play:{url}[;loop={yes|no}][;p ause={time-in-ms}][;total={time-i n-ms}]` or `Play:{url}[;loop={yes|no}][;pause ={time-in-ms}][;total={time-in-ms }]`	Plays the wave or MP3 file at the given `{url}`. The `{url}` must start with `http://`, `https://`, or `file://`. For the last case, the file name must be a valid internal path such as `/data/ringtones/Office A.wav`. Three semicolon-separated parameters may be added after the URI to control looping and pause between repetitions: `;loop=yes | no` (default is `no`) `;pause={time-in-ms}` (default is `0`) `;total={time-in-ms}` (default = `0` — play forever) For example: `Play:http://ringtones.com/beep.wav;loop =yes;pause=4000`	Y
`Wav.Stop:{url}`	Stops the playing of the wave or MP3 file at the given `{url}`	Y
`QueryStringParam:{query-string}`	A query string to append to the URI when the URI of a selected item is fired	
`Install:Select`	Executes the `<InstallURI>` element of the current highlighted menu item. If there is no highlighted item, or if the item does not have a valid `<InstallURI>` element, this action is ignored.	Y
`Install:{target}={url}`	Download the data at the given `{url}` and store in the internal `{target}` path. If the `{target}` is not a valid internal path, this action is ignored. For example: `Install:/backgnd/CherryBlossom.jpg= http://abc.com/pics/cherryBlossom.j pg` For a list of valid `{target}`, see the description of <InstallURI> element.	Y
`HangUp`	Hang up the phone by turning off the current audio device (including handset). All active calls end.	Y

Action URI Values

Action URI	Description	Allow Exit Attribute
`OBiStatus`	Trigger a full obi-status NOTIFY sent to the configured URLs.	Y
`End:{call-ref}`	End the call with the given `call-ref`.	Y
`Answer:{call-ref}`	End the call with the given `call-ref`. Any other call that is connected is placed on hold. Calls that are not yet connected will end.	Y
`Hold:{call-ref}`	Hold the call with the given `call-ref`.	Y
`Resume:{call-ref}`	Resume the holding call with the given `call-ref`. All other connected calls are placed on hold. Calls not yet connected will end.	Y
`Add2Conf`	Resume the holding call with the given `call-ref`. Other calls continue on at their respective current states.	Y
`Forward:{call-ref}:{target-number}`	Forward the (ringing) call with the given `call-ref` to the given target number.	Y
`Transfer:{call-ref}:{target-call-ref}`	Transfer the call with the given `call-ref` to the target call with a different `call-ref`.	Y
`BlindTransfer:{call-ref}:{target-number}`	Blind Transfer the call with the given `call-ref` to the given target number.	Y
`DND:{0|1}`	Enable (`1`) or Disable (`0`) **Do Not Disturb** in user preferences.	Y
`DNR:{0|1}`	Enable (`1`) or Disable (`0`) **Do Not Ring** in user preferences.	Y
`CWA:{0|1}`	Enable (`1`) or Disable (`0`) **Call Waiting** in user preferences.	Y
`CFA:{0|1}`	Enable (`1`) or Disable (`0`) **Call Forward All** in user preferences.	Y
`BAC:{0|1}`	Enable (`1`) or Disable (`0`) **Block Anonymous Call** in user preferences.	Y
`BCI:{0|1}`	Enable (`1`) or Disable (`0`) **Block Caller ID** in user preferences.	Y
`PG1:{0|1}`	Enable (`1`) or Disable (`0`) **Page Group 1** in user preferences.	Y
`PG2:{0|1}`	Enable (`1`) or Disable (`0`) **Page Group 2**.	Y

Built-In Icons

The following icons are built into the phone, and can be presented by name instead of by a URL or an image file.

Built-In Icons

Name	Description	Icon
acd	ACD Agent Function	
acd-avail	ACD Agent in Available State	
acd-unavil	ACD Agent in Unavailable State	
acd-wrapup	ACD Agent in Wrapping Up (Last Call) State	
acd-signoff	ACD Agent signed off	
actionurl	Action URL function	
alert	Generic Alert Message	
handset	Handset	
headset	Headset	
rj9	RJ9 Headset	
speaker	Speaker	
dial	Dial	
bci	Block (Outgoing) Caller ID	

Built-In Icons

Name	Description	Icon
blank		
blf-busy	BLF – Monitored Extension is in a call	
blf-hold	BLF – Monitored Extension has a holding call	
blf-idle	BLF – Monitored Extension is idle (no calls)	
blf-offline	BLF – Monitored Extension is offline	
blf-parked	BLF – Monitored Extension has a call parked	
blf-ring	BLF – Monitored Extension is ringing	
blf-blue	BLF function	
blf-green	BLF function	
blf-grey	BLF function	
blf-purple	BLF function	
blf-red	BLF function	
brush		
cell-phone		
buddy-list	Buddy List function	

Built-In Icons

Name	Description	Icon
blind-xfer	Blind Transfer Function	
call-history	Call History	
caution	Generic Caution	
cfwd-blue	Call Forward Enabled	
cfwd-red	Call Forward Enabled	
check	Check box	
chis-all	All Calls Call History	
chis-missed	Missed Calls Call History	
chis-out	Outgoing Calls Call History	
chis-recv	Received Calls Call History	
conf	Conference	
question		
connected	Call Connected State	
connected-hd	Call Connected in HD State	
park-monitor	Call Park (Slot) Monitor Function	

Built-In Icons

Name	Description	Icon
call		
dir	Directory or Phone Book or Contacts	
dnd	Do Not Disturb	
dnr	Do Not Ring	
star	Star or Favorite	
folder		
hold	Call Holding State	
home	Home	
hotel-blue	Hoteling Function	
hotel-green	Hoteling Function	
hotel-grey	Hoteling Function	
hotel-purple	Hoteling Function	
hotel-red	Hoteling Function	
hour-glass		
intercom	Intercom function (1-to-1 paging)	

Built-In Icons

Name	Description	Icon
lock		
unlock		
tools		
mail	Voicemail	
network	LAN or Generic Network	
pref	User Preferences	
pres	Presence Application	
pres-unknown	Presence is Unknown	
pres-away	Presence is Away	
pres-dnd	Presence is DND (Busy)	
pres-offline	Presence is Offline	
pres-online	Presence is Online	
pres-xa	Presence is Extended Away	
prod	Product Information	
sidecar	Sidecar or Expansion Module	
settings	Settings	

Built-In Icons

Name	Description	Icon
speed-dial	Speed Dial	
speak1	Speaker 1	
speak2	Speaker 2	
speak3	Speaker 3	
usb	USB Dongle, Storage, and so forth.	

Monitor OBi Phone Status

An OBi phone can (SIP) NOTIFY one or more preconfigured recipients its full status on-demand or partial status-delta update automatically by setting up the **OBiStatusNotifyURL** configuration parameter (one instance per SP*n* Service that is also configured to use the SIP signaling protocol). The value should be a valid address or FQDN with optional user name field, where the SIP NOTIFY is sent by the phone (for example, `status@192.168.15.123:5089` or `status.server.abc.com`). Note that the same reports are sent to all recipients, regardless of SP*n*.

The SIP/NOTIFY event is obi-status with the Content-Type application/x-obi-status. The message body is an <obi-status> document as shown in the following examples:

```
<?xml version="1.0" encoding="UTF-8" ?>
<obi-status full="1" obi-number="500123456" ncalls="2" dnd="0" cfa="0" ver="100">
  <calls>
    <call state="connected" time="2015-5-4T17:5:31" dur="5" serv="SP5">
      <peer>
        <name/>
        <number>+14089991234</number>
      </peer>
    </call>
    <call state="holding" time="2015-5-4T17:5:11" dur="25" serv="SP6">
      <peer>
        <name>Samuel Donaldson</name>
        <number>+15109791354</number>
      </peer>
    </call>
  </calls>
  <sp1 status="up" ncalls="0" proto="SIP" user="obihai-user1" mwi="0"/>
  <sp2 status="nocfg"/>
  <sp3 status="disabled"/>
  <sp4 status="up" ncalls="0" proto="SIP" mwi="1"/>
```

```
    <sp5 status="up" ncalls="1" proto="GV" user="obihai-user1@gmail.com" mwi="0"/>
    <sp6 status="up" ncalls="2" proto="SIP" user="tester12345" mwi="0"/>
    <obitalk status="up"/>
</calls>
</obi-status>
<?xml version="1.0" encoding="UTF-8" ?>
<obi-status full="0" obi-number="500123456" ver="101">
    <calls full="1">
      <call state="holding" time="2015-5-4T17:5:31" dur="15" serv="SP5">
        <peer>
          <name/>
          <number>+14089991234</number>
        </peer>
      </call>
      <call state="connected" time="2015-5-4T17:5:11" dur="35" serv="SP6">
        <peer>
          <name>Samuel Donaldson</name>
          <number>+15109791354</number>
        </peer>
      </call>
    </calls>
</calls>
</obi-status>
```

Elements of <obi-status> Document

The following are the elements of an <obi-status> document.

- <obi-status>
- <calls>
- <call>
- <peer>
- <name>
- <number>
- <spn>, n = 1 – 6
- <obitalk>

<obi-status>

The root element.

Attributes:

- `full`: 0 or 1. 0 if this is a full status report, or 1 if this is a partial update. For partial updates, unspecified elements are unchanged since last report.
- `ver`: An unsigned integer that starts at 0 for the first report after boot up, then increments by 1 for each subsequent report. The recipient may use this information to detect if it has missed a partial update (and request a full update to resynchronize).

- `obn`: The 9-digit OBi number of the reporting device.
- `ncalls`: An unsigned integer representing the number of calls currently active/holding on the device.
- `nmissed`: Number of new missed calls.
- `ring`: 1 if phone is ringing (one or more incoming calls alerting), or 0 if not.
- `mwi`: 1 if phone has new messages waiting, or 0 if not.
- `dnd`: 0 or 1. 0 if the Phone-Level DND feature is disabled, or 1 if it is enabled.
- `cfa`: 0 or 1. 0 if the Phone-Level Call-Forward-All (also known as Call-Forward-Unconditional) is disabled, or 1 if it is enabled
- `bac`: 0 or 1. 0 if the Phone-Level Block Anonymous Call feature is disabled, or 1 if it is enabled.
- `bci`: 0 or 1. 0 if the Phone-Level Block Caller ID (also known as Anonymous Call) feature is disabled, or 1 if it is enabled.
- `cwa`: 0 or 1. 0 if Call Waiting is disabled, or 1 if it is enabled.
- `dnr`: 0 or 1. 0 if the Do-Not-Ring feature is disabled, or 1 if it is enabled.
- `pg1`: 0 or 1. 0 if the phone has left Page Group 1, or 1 if it has joined Page Group 1.
- `pg2`: 0 or 1. 0 if the phone has left Page Group 2, or 1 if it has joined Page Group 2.
- `audio`: The current audio path which can be one of the following values:
  - `none`: Audio path is turned off.
  - `speaker`: Speakerphone is turned on.
  - `headset-RJ9`: The RJ-9 Headset is turned on and selected.
  - `handset`: The handset is off-hook and is being used.

Contains:

- (Optional) `<calls>, <sp1>, <sp2>, … <sp6>, <obitalk>`

\<calls>

A child element of `<obi-status>`.

Attributes:

- `full`: 0 if this contains only the `<call>` that has been updated. An unlisted `<call>` can be assumed to have the same state. 1 if this contains all the `<call>` elements on the phone. A call that is not listed means it does not exist (for example, the call ended). Note that full="1" is implied here if this is a full obi-status report, otherwise full="0" is implied.

Contains:

- (Optional) One or more `<call>` elements.

\<call>

A child element of `<calls>`. It contains information about an on-going call on the phone.

Attributes:

- `id`: A reference ID to identify the call. The exact same ID must be specified when invoking the following Action URIs that apply to a call:
  - `End`: To end the call.

- ➢ `Hold`: To hold the call.
- ➢ `Resume`: To resume the call.
- ➢ `BlindTransfer`: To blind transfer the call to a target number.
- `start`: The start time of this call in the format: `year-month-dayThour:min:sec` such as `2015-5-4T17:35:10`. Before the call is answered, the start time marks the time when the call is created (that is, when the dial-tone starts playing for an outgoing call, or when the phone starts ringing for an incoming call). After the call is answered, the start time marks the time at which the call is answered (when the called party answers for an outgoing call, or when this phone answers for an incoming call).
- `dur`: An unsigned integer that represents the duration of the call. Before the call is answered, the duration counts the time since the call is created. After the call is answered, it counts the time since the call is answered.
- `state`: The call state, which can be one of the following values:
  - ➢ `dialtone`: Dial tone being played to prompt user to enter the number to call.
  - ➢ `dialing`: User entering a number to call.
  - ➢ `trying`: Waiting for an outgoing call to answer.
  - ➢ `peerring`: Called party is ringing.
  - ➢ `ring`: Incoming call ringing on this phone.
  - ➢ `connected`: Call is answered.
  - ➢ `holding`: Call is placed on hold.
- `serv`: The voice service used for the call. It may be one of the following values:
  - ➢ `{empty-string}`: No specific service assigned to the call yet. This could be the case during dialing when no service-specific call appearance key is used for the call.
  - ➢ SP*n*, *n* = 1 – 6.
  - ➢ OBiTALK.

\<peer\>

A child element of `<call>`. It contains information about the call peer.

Attributes: None.

Contains:

- (Optional) `<name>, <number>`.

\<name\>

A child element of `<peer>`.

Attributes: None.

Contains:

- The Caller-ID name of the call peer.

\<number\>

A child element of `<peer>`.

Attributes: None.

Contains:

- The Caller-ID number of the call peer.

<sp*n*>, *n* = 1 – 6

A child element of `<obi-status>`. It describes the status of the SP*n* Service.

Attributes:

- `proto`: The signaling protocol of this service, either "SIP" or "GV".
- `user`: The user name of the service account.
- `status`: The status of the SP*n* service.

    For SIP, `status` can take one of the following values:

    - ➢ `disabled`: The service is disabled by configuration.
    - ➢ `nocfg`: The service is not configured.
    - ➢ `up`: Service is functioning normally.
    - ➢ `down`: No service at the moment.
- `ncalls`: Number of calls currently using this service.
- `dnd`: 1 if the service-level Do Not Disturb feature is enabled on this service, or 0 if it is not.
- `cfa`: 1 if the service-level Call Forward All (also known as Call Forward Unconditional) feature is enabled on this service, or 0 if it is not.
- `bac`: 1 if the service-level Block Anonymous Call feature is enabled on this service, or 0 if it is not.
- `bci`: 1 if the service-level Block Caller ID (also known as Anonymous Call) feature is enabled on this service, or 0 if it is not.
- `mwi`: 1 indicates if there are new messages in the mailbox associated with this service, or 0 if there are none.

Contains: None.

<obitalk>

A child element of `<obi-status>`. It describes the status of the OBiTALK service on the phone.

Attributes:

- `status`: Status of this service, which can be one of the following values:
    - ➢ `acquiring-service`: Acquiring service from OBiTALK network.
    - ➢ `registering`: Registering with OBiTALK network.
    - ➢ `connecting`: Connecting to OBiTALK network when using TCP to connect.
    - ➢ `up`: Service is functioning normally.
    - ➢ `backing-off`: Waiting to re-establish service after a failure.
- `ncalls`: The number of calls on this phone currently using this service.
- `dnd`: 1 if the service-level Do Not Disturb feature is enabled on this service, or 0 if it is not.

- `cfa`: 1 if the service-level Call Forward All (also known as Call Forward Unconditional) feature is enabled on this service, or 0 if it is not.
- `bac`: 1 if the service-level Block Anonymous Call feature is enabled on this service, or 0 if it is not.
- `bci`: 1 if the service-level Block Caller ID (also known as Anonymous Call) feature is enabled on this service, or 0 if it is not.

Contains: None