Grandstream Networks, Inc.

GXP21xx XML Application Guide

Version 1.0.0.2

GXP21xx XML Application Guide

Index

INTRODUCTION

The Grandstream GXP21xx supports XML Application on firmware version 1.0.3.30/1.0.4.x. This XML API allows users to access service information in web server, present the information in LCD as well as customize screen display. It's a custom application where the XML framework is an interactive, real-time implementation and XML message is dynamic, depending on a configurable object set. It could be easily developed with all the programming languages for web application development (e.g., PHP, ASP, and etc). Typical XML applications could be call center service, survey, reservations and much more.

REVISION HISTORY

VERSION 1.0.0.1

- Initial version for GXP21xx firmware version 1.0.3.30.

VERSION 1.0.0.2

- Updated for GXP21xx firmware version 1.0.4.23.
- Added HTTPS support in XML application, for both XML application server path and Softkey action.
- The allowed XML Server Path length is extended from 128 characters to 256 characters.
- Element <LeftStatusBar> is removed from template.xml. This element is optional. If present, it will override the default LeftStatusBar display pre-defined on the phone.
- Element <SoftkeyBar> is removed from template.xml. This element is optional. If present, it will override the default SoftkeyBar display pre-defined on the phone.
- Added Radio button support by using type="radio" in <input> element. Supported attributes are "group", "value", "selected" and "label". When the query is sent back the result is group=value. For example, a radio with group "answerToQ1" with a value of "B" would result in "answerToQ1=B".
- Added Checkbox support by using type="checkbox" in <input> element. Supported attributes are "name", "value" and "label". When the query is sent back the result is name=value.
- Added Selection list support by using <select> element.
- Added "password" type in <input> element so the entered digits will be displayed as *.
- Added "hidden" type in <input> element so the input filed and entered digits won't display. It can be used to pass value which is preferred to be hidden on LCD.
- Added system variable $xU for account x' s SIP User ID to be used in XML application where x is 0 to 5 for account 1 to account 6.
- Added system variable $xN for account x' s Display Name if filled in, otherwise account x' s Account Name (x is 0 to 5 for account 1 to account 6).
- Added system variable $xS for account x' s SIP Server where x is 0 to 5 for account 1 to account 6.

WHAT IS XML

XML (eXtensible Markup Language) is a markup language* for documents and applications containing structured information. This information contains both content (text, pictures, input box and etc.) and an indication of what role that content plays (e.g. content in a section header is different from content in a footnote, or content in a figure caption, or content in a database table, etc.). Almost all documents have certain kind of structure.

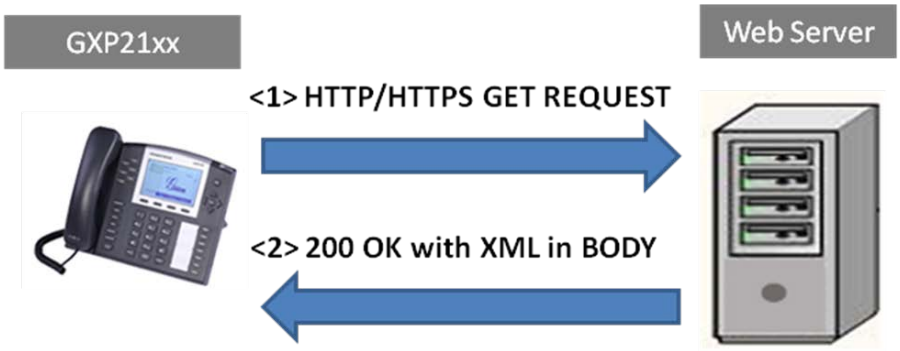
*A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents.

WHY XML

What benefits does XML provide to SIP endpoints? XML enables our SIP phones to serve as output devices for web services and other online applications. The XML infrastructure allows the phones to interact with external applications in a flexible and programmable manner. Three specific XML API supported by GXP21xx are XML Custom Screen, XML Phonebook and XML Application.

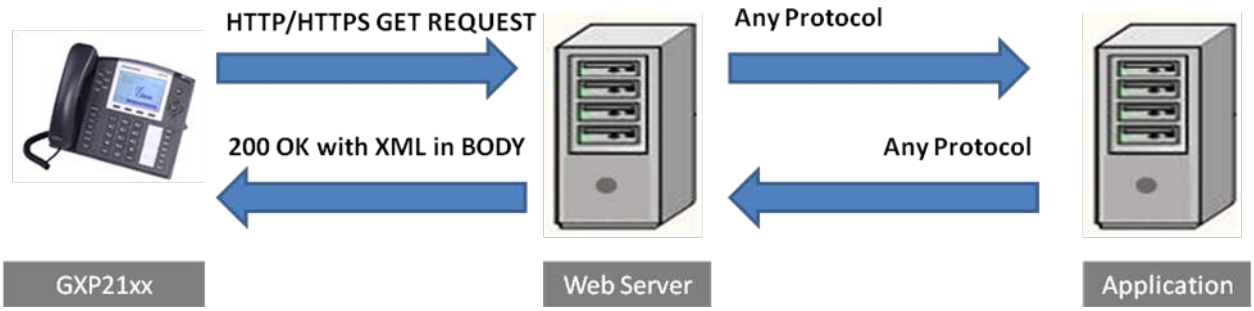
XML API ARCHITECTURE

The XML application on GXP21xx is using HTTP/HTTPS as the transport protocol. The following figure shows how it works. Basically, GXP21xx initiates the HTTP/HTTPS GET Request to the server and waits for the response. Once the phone receives the response with XML content in BODY, it displays the information.

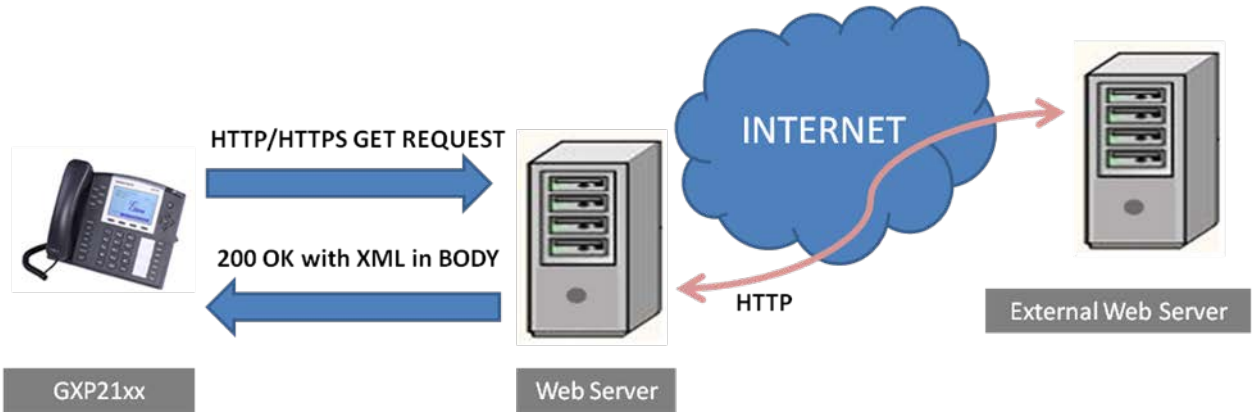


Two types of XML API architecture are introduced below, depending on if the application is within LAN or accessed via Internet.

1. An application in LAN area may exchange information in the following manner. GXP21xx sends request and accepts XML contents via HTTP/HTTPS, directly communicating with the server. The server will then handle the request and response via any protocols with the other application server to get the expected information for the XML service.



2. If the above Web Server accesses Internet, it could interact with outside web server and respond real-time content to GXP21xx.



As illustrated above, all of the application logic lies within the server side of the architecture. This allows faster applications development and minimal phone side maintenance.

XML APPLICATION CONFIGURATION

FIRMWARE

Before the XML application is launched on GXP21xx, the firmware on the phone must be upgraded to version 1.0.3.30/1.0.4.x. This document describes functions and features based on firmware version 1.0.4.23. Please refer to the following link for firmware upgrading information:

http://www.grandstream.com/support/firmware

WEB GUI CONFIGURATION

To launch the XML application on GXP21xx, firstly set the XML application server path and softkey label in web GUI->**Settings**->**Advanced Settings** page.

The Server Path must be set to the web server directory where the application is located. Currently HTTP and HTTPS are supported. The accepted format are as below (content in [ ] is optional).

- **For HTTP**

IP_address[:port]/dir/filename or ***http://IP_address[:port]/dir/filename***
Hostname[:port]/dir/filename or ***http://Hostname[:port]/dir/filename***

- **For HTTPS**

https://IP_address[:port]/dir/filename
https://Hostname[:port]/dir/filename
https://username:password@IP_address[:port]/dir/filename
https://username:password@Hostname[:port]/dir/filename

- **Examples:**

http://192.168.40.10/XMLapp/welcome.xml
mycompany.com:8080/surveyapp.xml
https://service.mycompany.com/XMLapp/reservation.xml
https://admin:adminpassword@192.168.40.123/GXP2120/API.xml

Note:

- If "http://" is not specified in the Server Path, the phone will use HTTP by default. Therefore, "http://" is not necessary to be included in the path if HTTP is used.
- For HTTPS, users must specify "https://" in the Server Path. If username and password are required by the HTTPS server, users could use "https://username:password@server_address/directory" to access.
- The content in [ ] is optional. If ":port" is not specified, by default port 80 will be used for HTTP and port 443 will be used for HTTPS.

The default value for Softkey Label is "XML Service". Users presses this softkey to launch the XML application. Please make sure the label string does not exceed the softkey maximum length on each model.

After the above configuration, click on "Update" in web GUI->**Settings**->**Advanced Settings** page. The GXP21xx will then display the XML Service softkey on LCD. Press the softkey with the configured label to launch the application.

Users may also use config file to provision the XML Application Server Path and Softkey Label to the phone. In this case, GXP21xx needs to be provisioned and rebooted. The corresponding P values are as below.

- P337: Server Path.
  This is a string of up to 256 characters that should contain the path to the XML application file.
- P352: Softkey Label.
  This is a string of up to 16 characters to display the softkey label on LCD.

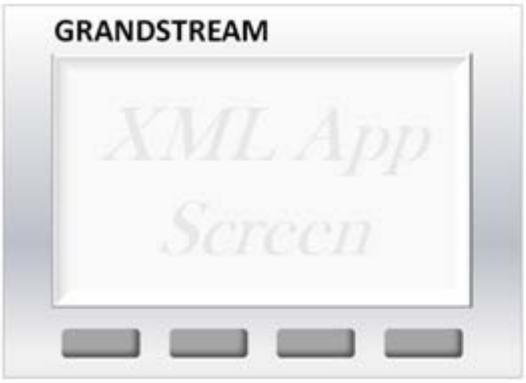
USING LCD AND KEYPAD

Without configuration, the default idle screen (GXP2120) is like below.

Once the XML Application server path and softkey label are successfully configured, the label will be displayed as the second softkey. Press this key to load the application.

As you may know, it is also possible to enter the application server path into PC's web browser. In this way you'll be able to see the exact XML document on your PC that your phone is receiving.

An example application screen will be like below.



NAVIGATION IN XML APPLICATION

Now the XML application is started on the phone. To navigate and proceed in the application, use the keypad and softkeys to complete the following different actions.

- Keypad digit input
  - Enter the digits on the phone's keypad if the application requires input.
  - To delete an input digit, press "Left" arrow key on the phone as backspace.
  - If there are multiple input fields, press "UP" or "DOWN" arrow to toggle among the input fields.
  - For example, enter the digits for **Your Applicant ID** in the input field as the above figure shows.

- Radio button selection
  - Press "UP" or "DOWN" arrow to move the selection cursor to the desired radio button.
  - Press "MENU" to make the selection.

- Checkbox selection
  - Press "UP" or "DOWN" arrow to move the selection cursor to the desired checkbox.
  - Press "MENU" to make the selection.

- Selection list
  - Press "UP" or "DOWN" arrow to move the selection cursor to the desired list field .
  - Press "LEFT" or "RIGHT" to toggle among different selections defined in the list.

- Softkey navigation
  - Press the corresponding softkeys to navigate. The softkeys are defined in the XML code. Each softkey could contain label for display, action and URL to trigger the phone to display another page. In the above example, "Submit" will navigate you to the next step of the screening process. "Exit" allows you to quit the application. "Tutorial" will launch the information page for users to browse.

- In the case when there are more than 3 or 4 softkeys (depending on the GXP21xx model) used in the XML application, a "MORE" softkey will automatically show. Click on it to view more softkeys.

- Quit application.
  - During the XML application, there is a default way for users to quit and return to normal idle screen. To quit, press and hold the "HOLD" button on phone's keypad for about 3 seconds.
  - A warning window will then be popped out "Quit Application: Are you sure you want to exit the current application?".
  - Press softkey "OK" to quit, or press "Cancel" to continue the application.
  - This operation is not applicable for GXP2124 yet on firmware version 1.0.4.23.

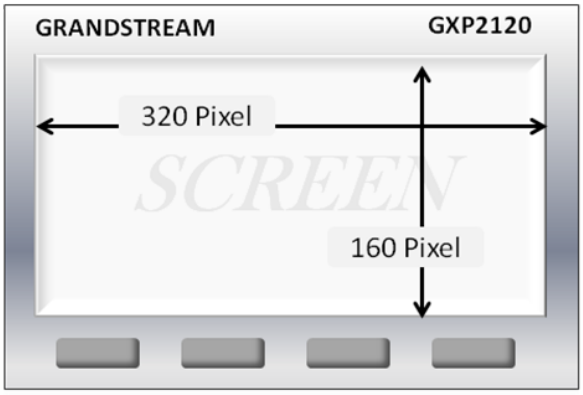
- Make/Answer call during application.
  - When the phone is running the application, users could make or answer call without exit.
  - Make call by offhook the handset, or press the speaker button.
  - Answer incoming call by offhook the handset, or press the speaker button, or press the softkey "AnswerCall".
  - Once the call screen is triggered, "ResumeAPP" softkey will display. Press this key to continue the application.

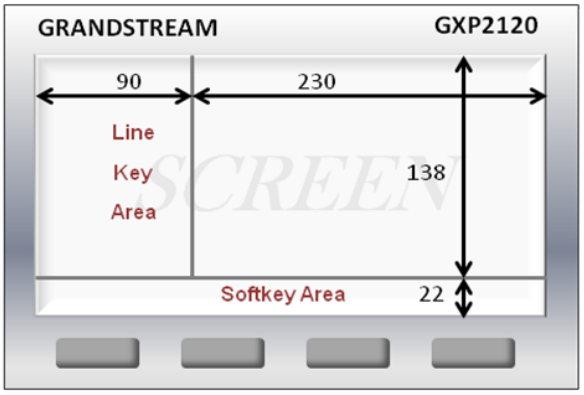
Note:

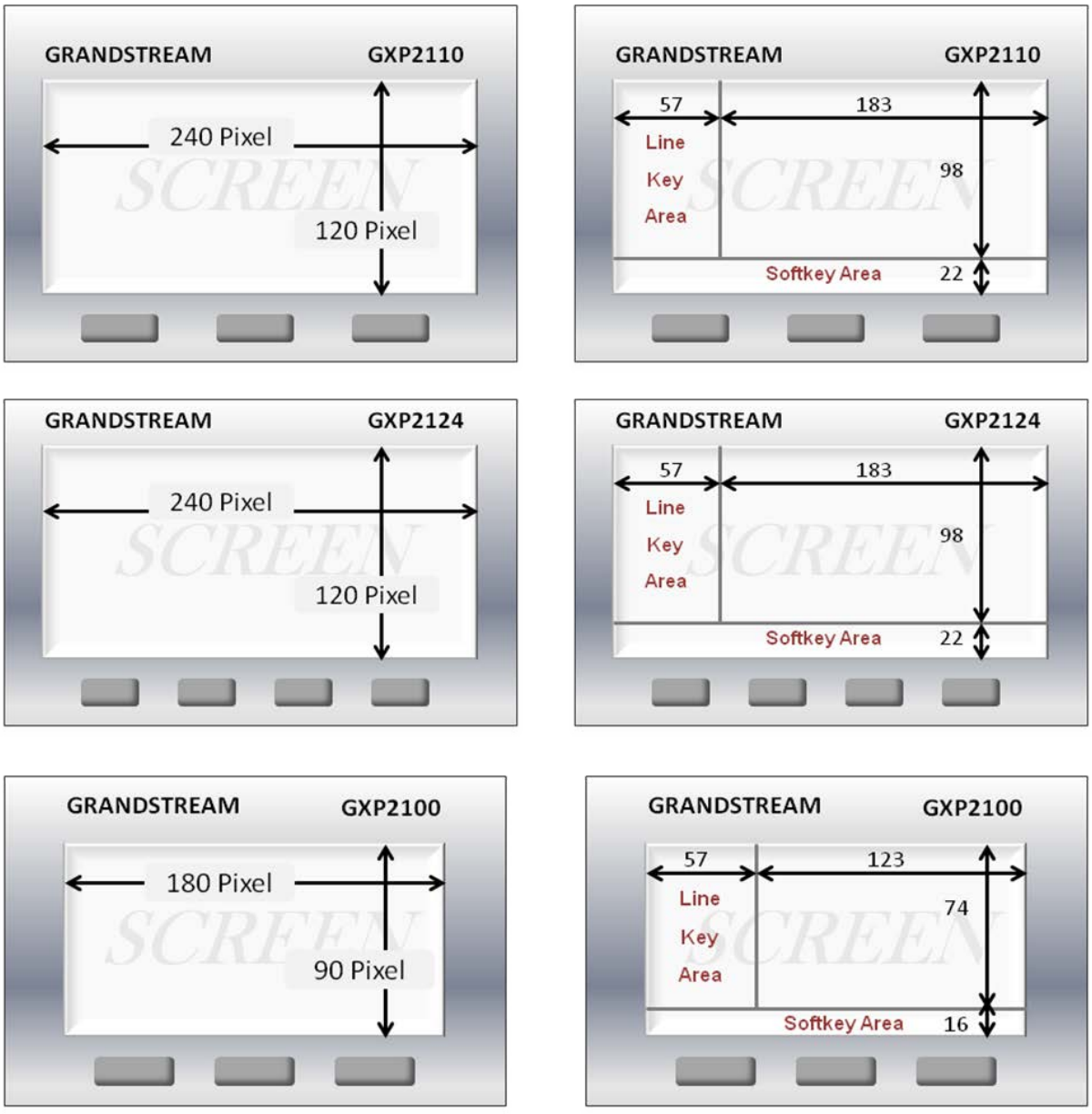
The above example and screenshots are taken on GXP2120. The configurations and operation are the same for all the other GXP21xx models, but the screen size and softkey layout may vary.

SCREEN LAYOUT

The GXP2120/2110/2100/2124 have different screen size and softkey layout. The LCD size (in pixel) and default layout are described below as reference.







Note:

Please refer to the actual device for the precise appearance and size.

XML APPLICATION FORMAT

HEADER

In the first line of the XML document, the following header can be set as XML declaration. It defines the XML version and encoding. On GXP21xx, UTF-8 is used as encoding method for correct display.

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

SPECIAL CHARACTERS

As followed by the standard XML recommendation, some characters need to be escaped. The following table lists the characters with their escape sequence.

Characters	Name	Escape Sequence
&	Ampersand	&amp
“	Quote	&quot
‘	Apostrophe	&apos
<	Left Angle Bracket	<
>	Right Angle Bracket	>

GXP21xx SCREEN XML STRUCTURE

The XML application template could be represented like below. <Page> section (mandatory) and <Events> section (optional) are the main customization area for the XML application. Please refer to the template.xml included in the package for more comments.

```
<Screen>
    <Page>
        <ShowStatusLine>false</ShowStatusLine>
        <Contents>
          <!-- ADD DISPLAY AND INPUT ELEMENTS FOR XML APPLICATION CONTENTS-->
        </Contents>
        <SoftKeys>
          <!-- ADD SOFTKEY ELEMENTS FOR XML APPLICATION CONTENTS-->
        </SoftKeys>
    </Page>

    <Events>
      <!--ADD EVENT HERE-->
    </Events>
</Screen>
```

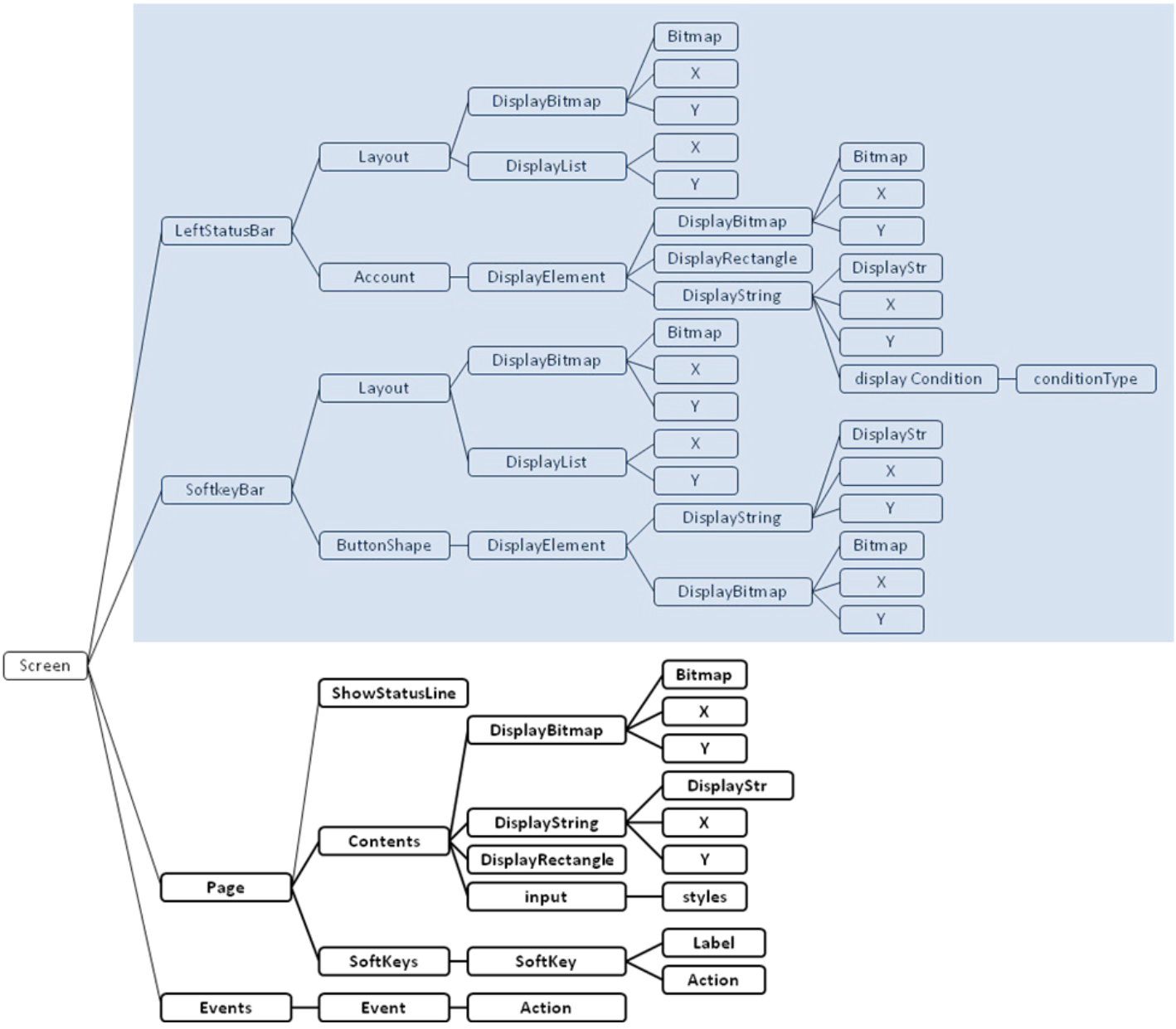
The XML document structure can be presented in the following diagram. This provides users an overview of the XML element and screen layout.

When creating your own XML document, it is not recommended to customize the <LeftStatusBar> section and <SoftkeyBar> section as shown in blue background of the following diagram. The default content for <LeftStatusBar> and <SoftkeyBar> are pre-defined for each model already. Therefore, unless users would like to override the default display on purpose, please do not include those two elements in the XML file. If needed, the default content for these two elements could be found in the same section of the XML customized idle screen templates. Please make sure use the correct templates for different models since they may have different width or height defined.

The XML customized idle screen templates can be downloaded in the following link:

http://www.grandstream.com/products/gxp_series/general/documents/GXP21xx_14xx_XML_Screen_Customization.zip

For the element attribute and text information, please refer to the details in the next section “GXP21xx XML ELEMENT”.


Screen
LeftStatusBar
Layout
DisplayBitmap
Bitmap
X
Y
DisplayList
X
Y
Account
DisplayElement
DisplayBitmap
Bitmap
X
Y
DisplayRectangle
DisplayString
DisplayStr
X
Y
display Condition
conditionType
SoftkeyBar
Layout
DisplayBitmap
Bitmap
X
Y
DisplayList
X
Y
ButtonShape
DisplayElement
DisplayString
DisplayStr
X
Y
DisplayBitmap
Bitmap
X
Y
Page
ShowStatusLine
Contents
DisplayBitmap
Bitmap
X
Y
DisplayString
DisplayStr
X
Y
DisplayRectangle
input
styles
SoftKeys
SoftKey
Label
Action
Events
Event
Action

GXP21xx XML ELEMENT

This section describes details of the XML element used in GXP21xx XML application. Please note that the element name is case-sensitive when being used in XML document.

ROOT ELEMENT <Screen>

<Screen> is the root element of the XML document used for GXP21xx XML application. This element is mandatory.

```
<Screen>
All the information for screen display is here
</Screen>
```

<Screen> ELEMENT DETAILS

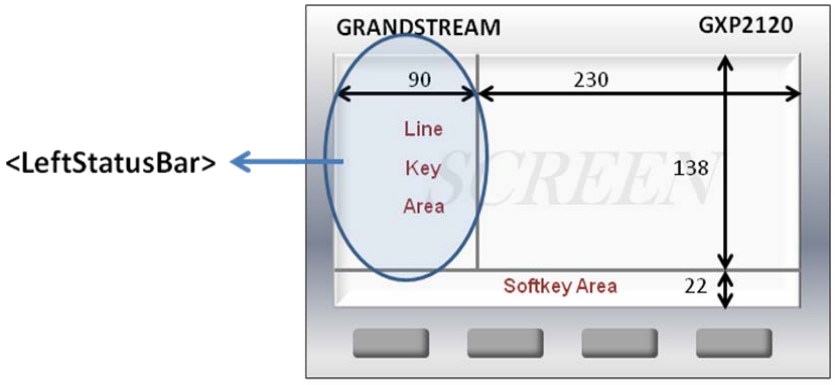
The following table shows child elements and attributes for <Screen> element.

Object	Position	Type	Values	Comments
Screen	**Root element**	**Mandatory**	**-**	**Root element of the XML document**
LeftStatusBar	Child element	Optional	-	Defines account bar display (customization not recommended)
SoftkeyBar	Child element	Optional	-	Defines softkey bar display (customization not recommended)
Page	Child element	Mandatory	-	Main customization area for XML application content and softkeys
Events	Child element	Optional	-	Used when certain event is triggered

<LeftStatusBar> ELEMENT

The <LeftStatusBar> section is used for displaying the account status and information in idle screen, i.e., the account name, registered or not, the background images for each bar. This element is optional. The size and format for each account are pre-defined for each model already and **we recommend users not to include this element** when creating the XML application.

Each model might have different width value defined for the LeftStatusBar. The following figure shows the <LeftStatusBar> area for GXP2120.




```
<LeftStatusBar>
  <Layout width="Width of the background layout">
    <DisplayList>
      <X>X location</X>
      <Y>Y location</Y>
    </DisplayList>
    <DisplayBitmap> Image information here </DisplayBitmap>
    <DisplayString> String information here</DisplayString>
  </Layout>
  <Account height="Height of the account display ">
    <DisplayElement>
      <DisplayBitmap> Image information here </DisplayBitmap>
    </DisplayElement>
  </Account>
</LeftStatusBar>
```

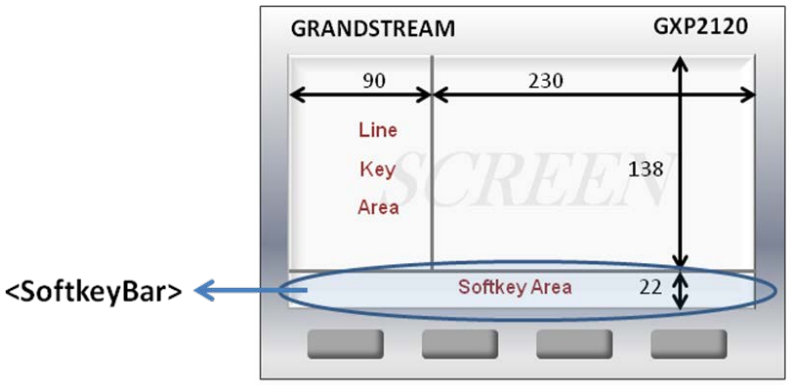
<LeftStatusBar> ELEMENT DETAILS

Object	Position	Type	Values	Comments
LeftStatusBar	**Element**	**Optional**	**-**	**Defines account bar display**
Layout	Child element	Mandatory	-	Defines account bar layout
width	<Layout> attribute	Mandatory	int	
Account	Child element	Optional	-	Defines display per account and it applies the same display to each account
height	<Account> attribute	Mandatory	int	

<SoftkeyBar> ELEMENT

The <SoftkeyBar> section defines how the softkey layout is, e.g, softkey bar height, position and button shape. This element is optional. The size and format are pre-defined for each model already and **we recommend users not to include this element** when creating the XML application.

Each model might have different height value defined for the Softkey bar. The following figure shows the <SoftkeyBar> area for GXP2120.




```
<SoftkeyBar>
    <Layout height="Height of the softkeyBar" >
        <DisplayBitmap>
            Image information here
        </DisplayBitmap>
        <DisplayList>
            <X>X location</X>
            <Y>Y location </Y>
        </DisplayList>
    </Layout>
    <ButtonShape id="Id number" width="Width of the button" height="Height of the button">
        <DisplayElement>
            DisplayElement information here
        </DisplayElement>
    </ButtonShape>
</SoftkeyBar>
```

<SoftkeyBar> ELEMENT DETAILS

Object	Position	Type	Values	Comments
SoftkkeyBar	**Element**	**Mandatory**	**-**	**Defines softkey bar display**
Layout	Child element	Mandatory	-	Defines softkey bar layout
height	<Layout> Attribute	Mandatory	int	
Buttonshape	Child element	Mandatory	-	Defines display per softkey and it applies the same display to each key
id	<Buttonshape> Attribute	Mandatory	int	
width	<Buttonshape> Attribute	Mandatory	int	
height	<Buttonshape> Attribute	Mandatory	int	

<Page> ELEMENT

This is the main customization section for the XML application.

```
<Page ignoreCallUpdate="true/false">
    <ShowStatusLine>false</ShowStatusLine>
    <Contents>
      Contents
    </Contents>
    <SoftKeys>
      Softkey
  </SoftKeys>
</Page>
```

<Page> ELEMENET DETAILS

Object	Position	Type	Values	Comments
Page	**Element**	**Mandatory**	**-**	
ignoreCallUpdate	<Page> Attribute	Optional	"true"/ "false"	When it's set to true, phone will not refresh the XML application screen to call screen for call status update (except the one triggered by pressing LINE key). Default value is false.
Contents	Child element	Mandatory	-	Defines all the strings, pictures, input,

				select fields
Softkeys	Child element	Mandatory	-	Defines softkey display and action.
ShowStatusLine	Child element	Optional	"true"/ "false"	It could use "true" or "false" as its text. "true": the line label on the left side will be always displayed during XML application. "false": the line label on the left side will not be displayed during XML application. So the XML application information could be shown in a full screen manner. Default is "true".

<Contents> ELEMENT

This is the main customization section for the XML application. A sample could be like below.

```
<Contents>
    <DisplayString>String information</DisplayString>
    <DisplayBitmap>Image information</DisplayBitmap>
    <DisplayRectangle x="X location" y="Y location" width="Width" height="Height"
bgcolor="Background color"/>
    <input name="input name" value="value" type="text" maxlength="max length" data-type="int">
    Input information
    </input>
    <select name="list name">
    Select information
    </select>
</Contents>
```

<Contents> ELEMENET DETAILS

Object	Position	Type	Values	Comments
Contents	**Element**	**Mandatory**	**-**	
DisplayString	Child element	Optional	-	Displays string
DisplayBitmap	Child element	Optional	-	Displays bitmap picture

DisplayRectangle	Child element	Optional	-	Displays rectangle
x	<DisplayRectangle> Attribute	Optional	int	Default value is 0
y	<DisplayRectangle> Attribute	Optional	int	Default value is 0
width	<DisplayRectangle> Attribute	Mandatory	int	
height	<DisplayRectangle> Attribute	Mandatory	Int	
bgcolor	<DisplayRectangle> Attribute	Optional	string	Default value is Black
input	Child element	Optional	-	Display input field
select	Child element	Optional	-	Display selection list field

<DisplayString> ELEMENT

This element is used for displaying string information on the screen.

```
<DisplayString font = "unifont" width="width of the string" height="height of the string" halign="
center/left/right" color="color of the string" bgcolor="color of the background" >
    <X>X location</X>
    <Y>Y location </Y>
    <DisplayStr>Display String</DisplayStr>
</DisplayString>
```

<DisplayString> ELEMENET DETAILS

Object	Position	Type	Values	Comments
DisplayString	**Element**	**Optional**	**-**	
font	<DisplayString> Attribute	Optional	"unifont"	"unifont" is supported and default
width	<DisplayString> Attribute	Optional	int	
height	<DisplayString> Attribute	Optional	int	
halign	<DisplayString> Attribute	Optional	string	Default value is "left"
color	<DisplayString> Attribute	Optional	string	Default value is "Black"
bgcolor	<DisplayString> Attribute	Optional	string	Default value is "White"
X	Child element	Mandatory	int	Displays the string from X
Y	Child element	Mandatory	int	Displays the string from Y
DisplayStr	Child element	Mandatory	string	The string to be displayed

<DisplayBitmap> ELEMENT

This element is to display a bitmap picture on the screen. Inside the <Bitmap> tag of the XML document, the picture must be encoded in base64 format already. There are plenty of base64 encoder online provided for encoding the .bmp picture. Please make sure the original .bmp picture is in monochrome grey level 8 before encoding.

```
<DisplayBitmap isfile="true/false" isflash="true/false">
    <Bitmap> Bitmap file encoded in base64 format </Bitmap>
    <X> X location </X>
    <Y> Y location </Y>
</DisplayBitmap>
```

<DisplayBitmap> ELEMENET DETAILS

Object	Position	Type	Values	Comments
DisplayBitmap	**Element**	**Optional**	**-**	
isfile	<DisplayBitmap> Attribute	Optional	"true"/ "false"	"true": to display the picture embedded in the firmware. Users won't be able to directly use it for customized pictures. "false": to display the picture customized in <Bitmap> tag. Default value is "false".
isflash	<DisplayBitmap> Attribute	Optional	"true"/ "false"	Default value is "false"
X	Child element	Mandatory	int	Displays the picture from X
Y	Child element	Mandatory	int	Displays the picture from Y
Bitmap	Child element	Mandatory	string	The base-64 encoded .bmp file

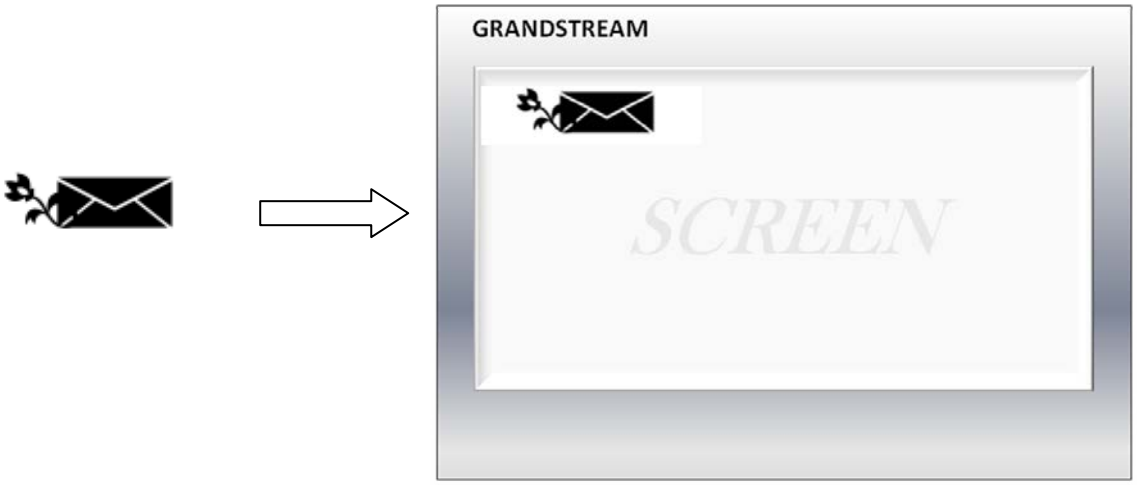
To create .bmp and display it on the phone:

- Firstly, make sure the picture is in .bmp format and not exceed the LCD size of the phone. GXP21xx support .bmp file with monochrome grey level 8 to be displayed.
- Use a base-64 encoder to encode the picture. Users shall find base64 encoders online.
- Copy and paste and encoded result inside <Bitmap> tag.

Example:

<DisplayBitmap isfile="false">
<Bitmap>Qk1EFgAAAAAAADYEAAAoAAAAggAAACMAAAABAAgAAAAAAA4SAAASCwAAEgsAAAAAAAAAAAAA
AAAAAAEBAQACAgIAAwMDAAQEBAAFBQUABgYGAAcHBwAICAgACQkJAAoKCgALCwsADAwMAA0NDQAODg4ADw8PA
BAQEAAREREAEhISABMTEwAUFBQAFRUVABYWFgAXFxcAGBgYABkZGQAaGhoAGxsbABwcHAAdHR0AHh4eAB8fHwAgI
CAAISEhACIiIgAjIyMAJCQkACUlJQAmJiYAJycnACgoKAApKSkAKioqACsrKwAsLCwALS0tAC4uLgAvLy8AMDAwADExMQAy
MjIAMzMzADQ0NAA1NTUANjY2ADc3NwA4ODgAOTk5ADo6OgA7OzsAPDw8AD09PQA+Pj4APz8/AEBAQABBQUEAQkJCA
ENDQwBEREQARUVFAEZGRgBHR0cASEhIAElJSQBKSkoAS0tLAExMTABNTU0ATk5OAE9PTwBQUFAAUVFRAFJSUgBTU
1MAVFRUAFVVVQBWVlYAV1dXAFhYWABZWVkAWlpaAFtbWwBcXFwAXV1dAF5eXgBfX18AYGBgAGFhYQBiYmIAY2NjAG
RkZABlZWUAZmZmAGdnZwBoaGgAaWlpAGpqagBra2sAbGxsAG1tbQBubm4Ab29vAHBwcABxcXEAcnJyAHNzcwB0dHQAd
XV1AHZ2dgB3d3cAeHh4AHl5eQB6enoAe3t7AHx8fAB9fX0Afn5+AH9/fwCAgIAAgYGBAIKCggCDg4MAhISEAIWFhQCGhoYA
h4eHAIiIiACJiYkAioqKAIuLiwCMjIwAjY2NAI6OjgCPj48AkJCQAJGRkQCSkpIAk5OTAJSUlACVlZUAlpaWAJeXlwCYmJgAmZm
ZAJqamgCbm5sAnJycAJ2dnQCenp4An5+fAKCgoAChoaEAoqKiAKOjowCkpKQApaWlAKampgCnp6cAqKioAKmpqQCqqqoAq
6urAKysrACtra0Arq6uAK+vrwCwsLAAsbGxALKysgCzs7MAtLS0ALW1tQC2trYAt7e3ALi4uAC5ubkAurq6ALu7uwC8vLwAvb29
AL6+vgC/v78AwMDAAMHBwQDCwsIAw8PDAMTExADFxcUAxsbGAMfHxwDIyMgAycnJAMrKygDLy8sAzMzMAM3NzQDOzs4
Az8/PANDQ0ADR0dEA0tLSANPT0wDU1NQA1dXVANbW1gDX19cA2NjYANnZ2QDa2toA29vbANzc3ADd3d0A3t7eAN/f3wDg
4OAA4eHhAOLi4gDj4+MA5OTkAOXl5QDm5uYA5+fnAOjo6ADp6ekA6urqAOvr6wDs7OwA7e3tAO7u7gDv7+8A8PDwAPHx8Q
Dy8vIA8/PzAPT09AD19fUA9vb2APf39wD4+PgA+fn5APr6+gD7+/sA/Pz8AP39/QD+/v4A////AP///
///8AAP///
///8AAP///
///8AAP///
///8AAP///
///////////////////////////////8AAP///
/////////8AAP//8AAP////
///OnF0aFSY/Y2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnbe
H/4f///////////////////////////////////8AAP//5/////////////////4cHDIOhGgEBAQEBAQEBAQEBAQEBAQEBAQ
EBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEZwPltav////////////////////////////////////8AAP//////////////////////////////////9P/P/wk//////////
2Rgc6ATHh6kABAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBATfh5z0Bav//////////////
////////////////////8AAP///////////////////////////////////////8Trv9SN////////xPCsNqAQETuvxwAQEBAQEBAQEBAQEBAQEBAQEBAQE
BAQEBAQEBAQEBAQEBAQEBAQEBZPnJHAEBav////////////////////////////////////8AAP//8lH4QBCuT//////3YH
vf9qAQEBBIr/pQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQeZ/5kHAQEBav////////////////////
///////////////8AAP///9JAQEBAVL8////nAGH//9qAQEBAQFVyAEBAQEBAQEBAQEBAQEBAQEBAQEBAQE
BAQEBAQEBAQEBAQEBAQEBHMb5ZwEBAQEBav/////////////////////////////////////8AAP//+TAQEBAQFMzPyo
Cjr8//9qAQEBAQEBBwEBLwEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQE95+Q3AQEBAQEBav/////////
//////////////////////////8AAP///5HwEBAQEBAQQfAZP///9qAQEBAQEBAQEh83kBAQEBAQEBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBAQEBAWr5wBkBAQEBAQEBav/////////////////////////////////////8AAP//2y4BAQE
BQ7cTH1j///9qAQEBAQEBAQEBgf+rDQEBAQEBAQEBAQEBAQFVXgEBAQEBAQEBAQEBAQEKn/+QBwEBAQEBAQEBav///
////////////////////////////////8AAP//Mmbfe/2oWcwfS//9qAQEBAQEBAQEBAUzz1SUBAQEBAQEBAQEBSc
z//9JSAQEBAQEBAQEBAR/M9l4BAQEBAQEBAQEBav//////////////////////////////////////8AAP//yQSNhAEi
3v9qAQEBAQEBAQEBAQEl1fBMAQEBAQEBAT3G/+ptZ+T/zEYBAQEBAQEBQOrhMQEBAQEBAQEBAQEBav//////////////////////
////////////8AAP//tHyv5mQEBE3xUAQEBAQEBAQEBAQEBDa7/fwEBAQE3uv/weQ0BAQpw6v/GP
QEBAQFz/L0TAQEBAQEBAQEBAQEBav////////////////////////////////////8AAP///zJ//////lMCs//yQEBAQEBAQEB
AQEBAQEBAQEBAQF5/LQNLrH/84EQAQEBAQEBDXnw/7o3CqX/igQBAQEBAQEBAQEBAQEBav////////////////////////////////////8AA
P//////////////////////////////vUMBATeN8/wHOr3/8EMHrv///BYBAQEJAQEBAQEBAQEBAQEBAQEBRu3q//aQEwEBAQEBAQEBAQE
QhPP/6vNYAQEBAQEBAQEBAQEBAQEBav////////////////////////////////////8AAP////////////////////////////9nAQEBAQEBE38NAQFPEC7D////
/4cBAQEFAQEBAQEBAQEBAQEBAQEin//5mRwBAQEBAQEBAQEBAQEBAROQ9v+rKwEBAQEBAQEBAQEBAQEBav//////////
/////////////////////////8AAP///////////////////////////20BAQEBAQEBAQEBAQEBrv////////xJAQEBAQEBAQEBAQEBAQEBGZb5/58iAQEBAQ
EBAQEBAQEBAQEBAQEBHJn5/6IlAQEBAQEBAQEBAQEBav/////////////////////////////////////8AAP///////////////////////////FUiAQEBAQFDyf/
hYQEB2P//////////mRABAQEBAQEBAQEBARON9v+rKwEBAQEBAQEBAQEBAQEBAQEBAQEBAQEin//8mRwBAQEBAQEBA
QEBav//////////////////////////////////////8AAP//////////////////////////////0j0BATf5////PQEB4f////////////xhAQEBAQEBAQEQgfP/tDQBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBASur//aTFgEBAQEBAQEBav/////////////////////////////////////8AAP//////////////////////////////1RYBAb3///lw
AR8E2P///////////95AQEBAQEBCnbt/8M6AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBMbT/84cQAQEBAQEB
av/////////////////////////////////////8AAP//////////////////////////////qGQEBAXaZjSgBAZbh9v////////////95AQEBAQdt5//JRgEBAQEBAQEBAQEBA
QEBAQEBAQEBAQEBAQEBAQEBAQE6w//wfA0BAQEBav////////////////////////////////////8AAP/////////////////////////////9PAQEBAQEB
AQEBAZ////////////////95AQEEZOH/0k8BAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAUPJ/+1wCg
EBav/////////////////////////////////////8AAP///////////////////////////7QBAQEBAQEBAQEBAbf///////////////95AVjY/9teAQEBAQEBAQEBAQEBAQ
EBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBT9L/5GcHav////////////////////////////////////8AAP///////////////////////////6VnQFKE
tzoBAQEBAef///////////////+xz//hZAQBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQ
Fb2P/esf/////////////////////////////////////8AAP////////////////////////////////////2oBAQEBQ///////////////////6m0HAQEBAQEBAQEBAQEBAQEBAQEB
AQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBBGTh///////////////////////////////////////8AAP/////////////////////////////////////1IBAQ
El3v////////////////2pZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZmZ
mZmf8////////////////////////////////////8AAP///////////////////////////////////8xMBFoHw///
///////////////////////////////////8AAP////////////////////////////////////5K7e//
//////////////////8AAP//8
AAAAA</Bitmap>
<X>0</X>
<Y>7</Y>
</DisplayBitmap>

The above example will render the following picture to be displayed on the top-left corner (X: 0, Y: 7) of the screen.



<input> ELEMENT

This element is to render input fields on screen so that users could enter necessary information to submit or proceed.

```
<input name="Input Name" value="Value" type="text" maxlength="Max Length" data-type="int">
    <styles pos_x="X location" pos_y="Y location" height="Height"/>
</input>
```

<input> ELEMENET DETAILS

Object	Position	Type	Values	Comments
input	**Element**	**Optional**	**-**	
name	<input> Attribute	Mandatory	string	A unique id for the input field
value	<input> Attribute	Optional	string	Input field value
type	<input> Attribute	Optional	"text" "password" "hidden" "radio" "checkbox"	"text": the input digits will display as it is; "password": the input digits will display as * after 2 seconds idle time or different key pressing; "hidden": the input digits will not display but the value can still be sent to the

				query. "radio": used for radio selection with attribute "group"; "checkbox": used for checkbox (multiple) selection (s).
maxlength	<input> Attribute	Optional	int	To limit the input length from users
data-type	<input> Attribute	Optional	"int"/"string"	Currently "int" or "string" can be used. Default value is "string"
styles	Child element	Mandatory	int	Input display style
pos_x	<styles> Attribute	Mandatory	int	Input field displays from X
pos_y	<styles> Attribute	Mandatory	int	Input field displays from Y
height	<styles> Attribute	Mandatory	int	Input field height

Example 1 - Input type "text":

An input field which requires maximum 4-digit number to login. The entered digit will display as it is.

```
<input name="memid" value="" type="text" maxlength="4" data-type="int">
    <styles pos_x="190" pos_y="85" border-color="Black" bgcolor="Light2" width="70"/>
</input>
```

Example 2 - Input type "password":

An input field which requires maximum 4-digit password to login. The entered digit will display as * after 2-second idle time or different key pressing.

```
<input name="memid" value="" type="password" maxlength="4" data-type="int">
    <styles pos_x="190" pos_y="85" border-color="Black" bgcolor="Light2" width="70"/>
</input>
```

Example 3 - Input type "hidden":

An input field which requires maximum 4-digit id to login. The input field, the pre-defined text in "value=" or the entered digits will not display. However, the value can still be passed to the query.

```
<input name="memid" value="" type="hidden" maxlength="4" data-type="int">
    <styles pos_x="190" pos_y="85" border-color="Black" bgcolor="Light2" width="70"/>
</input>
```

Example 4 - Radio button:

Two groups are defined in this example with value 1, 2, 3, 4 to be selected for each group. Only one radio button for each group could be selected.

```
<input group="group1" value ="4" type="radio" selected="1" label="">
        <styles pos_x="120" pos_y="35"/>
</input>
<input group="group1" value="3" type="radio" selected="0" label="">
        <styles pos_x="140" pos_y="35"/>
</input>
<input group="group1" value ="2" type="radio" selected="0" label="">
        <styles pos_x="160" pos_y="35"/>
</input>
<input group="group1" value="1" type="radio" selected="0" label="">
        <styles pos_x="180" pos_y="35"/>
</input>

<input group="group2" value ="4" type="radio" selected="1" label="">
        <styles pos_x="120" pos_y="50"/>
</input>
<input group="group2" value="3" type="radio" selected="0" label="">
        <styles pos_x="140" pos_y="50"/>
</input>
<input group="group2" value ="2" type="radio" selected="0" label="">
        <styles pos_x="160" pos_y="50"/>
</input>
<input group="group2" value="1" type="radio" selected="0" label="">
        <styles pos_x="180" pos_y="50"/>
</input>
```

Example 5 - Checkbox:

The following example shows four checkboxes to be selected. Users could do multiple selection by using checkbox.

```
<input name="checkbox1" value="Dog" type="checkbox" label="Dog" >
        <styles pos_x="5" pos_y="50"/>
</input>

<input name="checkbox2" value="Cat" type="checkbox" label="Cat" >
        <styles pos_x="5" pos_y="65"/>
</input>

<input name="checkbox3" value="Bird" type="checkbox" label="Bird" >
        <styles pos_x="5" pos_y="80"/>
</input>

<input name="checkbox4" value="Rabbit" type="checkbox" label="Rabbit" >
        <styles pos_x="5" pos_y="95"/>
</input>
```

<select> ELEMENT

This element is to render selection list fields on screen so that users could choose the answer to submit or proceed. "name=value" will be passed to the query. The text for <item> element is the displayed option for the list.

```
<select name="List Name">
    <styles pos_x="X location" pos_y="Y location" />
    <items>
        <item value="value 1">item 1 string</item>
        <item value="value 2">item 2 string</item>
        <item value="value 3">item 3 string</item>
        <item value="value 4">item 4 string</item>
     </items>
</select>
```

<select> ELEMENET DETAILS

Object	Position	Type	Values	Comments
select	**Element**	**Optional**	**-**	
name	<select> Attribute	Mandatory	string	A unique id for the select field
styles	Child element	Mandatory	-	Selection list display style
pos_x	<styles> Attribute	Mandatory	int	Select field displays from X
pos_y	<styles> Attribute	Mandatory	int	Select field displays from Y
items	Child element	Mandatory	-	Selection list options
item	<items> Child element	Mandatory	-	Attribute "value" is mandatory; Text is the displayed string for this option.

Example:

An selection list for age is defined.

```
<DisplayString>
        <X>5</X>
        <Y>65</Y>
<DisplayStr>Age</DisplayStr>
</DisplayString>

<select name="Age">
        <styles pos_x="5" pos_y="80"/>
                <items>
                        <item value="1">Under 25</item>
                        <item value="25">25 to 50</item>
                        <item value="50">Above 50</item>
                        <item value="100">Refused to answer</item>
                </items>
</select>
```

<SoftKeys> ELEMENT

This element is the parent element for <SoftKey> element. The purpose is to set up the softkey display and action. This element is mandatory.

```
<SoftKeys>
    <SoftKey>
    Softkey information here
    </SoftKey>
 </SoftKeys>
```

<SoftKeys> ELEMENET DETAILS

Object	Position	Type	Values	Comments
SoftKeys	**Element**	**Mandatory**	**-**	
softkey	Child element	Mandatory	-	Defines each softkey' s display and action

<SoftKey> ELEMENT

This element defines each softkey's label and action. This element is mandatory. If there are more than 3 or 4 softkeys set up here (depending on the softkey number of different models), a "MORE" softkey will be displayed automatically to access all the active softkeys.

```
<SoftKey action="Softkey action" label="Softkey label" commandArgs="Argument" />
```

<SoftKey> ELEMENET DETAILS

Object	Position	Type	Values	Comments
SoftKey	**Element**	**Mandatory**	**-**	
action	<SoftKey> Attribute	Mandatory	string	"Dial", "UseURL", "AppendInputURL", "QuitApp"
label	<SoftKey> Attribute	Mandatory	string	Displays the softkey name
commandArgs	<SoftKey> Attribute	Optional	string	URL information, or number
commandId	<SoftKey> Attribute	Optional	int	Only for action=Dial. This specifies the account index to dial out the call from, starting from 0 for account 1

Example 1 - Dial:

A softkey "Call" will be displayed. When pressing on it, 1087 will be dialed out using account 1.

```
<SoftKey action="Dial" label="Call" commandId="0" commandArgs="1087" />
```

Example 2 - UseURL:

A softkey "Start" will be displayed. When pressing it, phone will send out HTTP request for the link specified in commandArgs. The XML document in link "http://192.168.40.144/hotelSurvey/login.php" will be displayed. This could be used for directing to Next page, or back to Previous page in the XML application.

```
<SoftKey action="UseURL" label="Start" commandArgs="http://192.168.40.144/Survey/login.php" />
```

The URL in commandArgs could also have variables directly appended to reflect user's response in XML application. The following "Yes" softkey pressing will trigger request for link with answer=yes at the end.

```
<SoftKey action="UseURL" label="Yes"
commandArgs="http://192.168.40.144/Survey/question1.php?answer=yes" />
```

HTTPS URL can also be used.

```
<SoftKey action="UseURL" label="Yes"
commandArgs="https://user:password@192.168.40.144/Survey/xmlpage1.xml" />
```

Example 3 - AppendInputURL:

A softkey "Login" will be displayed. When pressing it, phone will send out HTTP request for the link with <input> element "name=value" appended at the end of the link. Then the corresponding page will be displayed. For example, if the <input> element has name=id, and the user enters "1234" in the screen input field, the requested link will be "http://192.168.40.144/hotelSurvey/login.php?id=1234". In this way, the input information could be passed to the next XML application pages.

```
<SoftKey action="AppendInputURL" label="Login"
commandArgs="http://192.168.40.144/Survey/start.php" />
```

If there are multiple input fields defined in one page, please make sure each input element has different name. And all of them will be appended to the commandArgs with the sequence of input fields.

Users could append the input field to a URL which contains the variable already. For example, user enters password 1234 in input field where input name=password. To construct link "192.168.40.144/Survey/page2.php?id=1234&password=1234", use the following commandArgs.

```
<SoftKey action="AppendInputURL" label="Next"
  commandArgs="192.168.40.144/Survey/page2.php?id=<?php echo $questionID; ?>" />
```

Also, HTTPS URL can be used.

```
<SoftKey action="AppendInputURL" label="Login"
commandArgs="https://192.168.40.144/Survey/start.php" />
```

Example 4 - QuitApp:

A softkey "Exit" will be displayed. When pressing it, phone will quit the XML application and go back to idle screen.

```
<SoftKey action="QuitApp" label="Exit" />
```

<Events> ELEMENT

This element enables the phone to respond to certain pre-defined event.

```
  <Events>
    <Event>
        Event
    </Event>
</Events>
```

<Events> ELEMENET DETAILS

Object	Position	Type	Values	Comments
Events	**Element**	**Optional**	**-**	
Event	Child element	Mandatory	-	Defines each event

<Event> ELEMENT

In one XML document, multiple <Event> elements could be used. However, the state for each event must be different so that the action is unique for that state.

```
<Event state="Event state">
        <Action action="Action name" commandArgs="Action argument"></Action>
</Event>
```

<Event> ELEMENET DETAILS

Object	Position	Type	Values	Comments
Event	**Element**	**Mandatory**	**-**	
state	<Event> Attribute	Mandatory	string	Please refer to "ATTRIBUTE state" section for values and meaning
Action	Child element	Mandatory	-	
action	<Action> Attribute	Mandatory	string	
commandArgs	<Action> Attribute	Optional	string	

Example:

When event "onhook" is triggered during the XML application, a HTTP request to the specified <URL> will be sent out. Then the screen will display the XML document in that link.

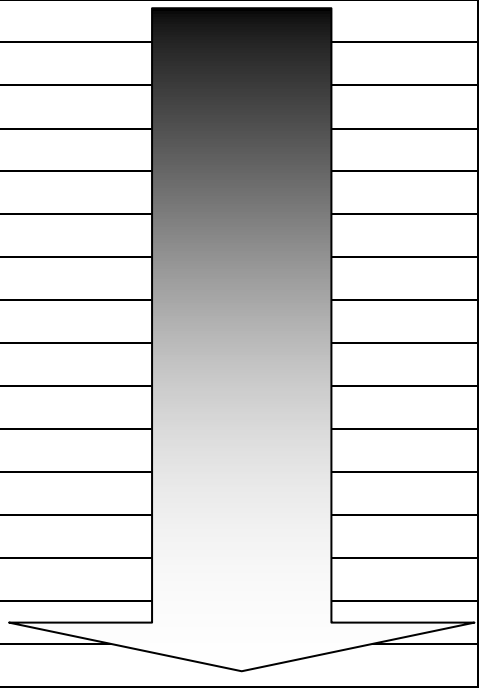
```
<Events>
        <Event state="onhook">
        <Action   action="UseURL" commandArgs="http://192.168.40.144/Survey/helpinfo.php"
        </Event>
</Events>
```

XML APPLICATION ELEMENT ATTRIBUTE

ATTRIBUTE color/bgcolor

- For "color" attribute, the default value is "Black".
- For "bgcolor" attribute, the default value is "White".
- For "border-color" attribute, the default value is "None".

color/bgcolor	Details
None	
Black	
Dark6	
Dark5	
Dark4	
Dark3	
Dark2	
Dark1	
Gray	
LightGray	
Light1	
Light2	
Light3	
Light4	
Light5	
Light6	
White	



ATTRIBUTE halign

Attribute halign is used for <DisplayString> element. The default value is "left".

halign	Details
center	GXP21xx XML Application
left	GXP21xx XML Application
right	GXP21xx XML Application

ATTRIBUTE state

Attribute state is used for <Event> element. This attribute is mandatory.

state	Details
callStateStarted	When phone is switching to call state (e.g., offhook to dialing state, or an incoming phone call) from idle
callStateEnded	When phone is switching to idle from call state
resumeFromCallEnded	Resume XML application from Call Ended state
resumeFromCallConnected	Resume XML application from Call Connected state
resumeFromCallFailed	Resume XML application from Call Failed state
resumeFromCallOnhold	Resume XML application from Call On Hold state
resumeFromCallRinging	Resume XML application from Call Ringing state
onhook	When phone is onhook
offhook	When phone is offhook

SYSTEM VARIABLES IN STRING DISPLAY

In <DisplayString> element, the following system variables could be used to display the pre-defined values in XML application (also applied to XML customized idle screen).

$String			
$a	This variable is replaced with the configured account name	$A	This variable is replaced with configured softkey label
$b	-	$B	This variable is replaced with the current day of month with leading zero, possible values: 01, 02, …, 31
$c	This variable is replaced with Missed Call string along with missed call count.	$C	This variable is replaced with DND (Do-Not-Disturb) label when DND is enabled
$d	This variable is replaced with the current day of month with leading zero, possible values: 1, 2, …, 31	$D	This variable is replaced with the current day of month with leading zero, possible values: 01, 02, …, 31
$e	This variable is replaced with the onhook dialing number	$E	-
$f	This variable is replaced with the Month-week-date format based on the configuration	$F	-
$g	This variable is replaced with the country name for weather information	$G	This variable is replaced with the number of the Missed Call
$h	This variable is replaced with the current hour of day in 12-hour format with leading zero, possible values: 01, 02, …, 12	$H	This variable is replaced with the current hour of day in 24-hour representation with leading zero, possible values: 00, 02, …, 23
$i	This variable is replaced with the system IPV6 Address	$I	This variable is replaced with the system IPV4 Address
$j	This variable is replaced with Forwarded Call string along with forwarded calls count	$J	-
$l	-	$L	This variable is replaced with the city name for weather information
$m	This variable is replaced with the current minute of hour with leading zero, possible values: 01, 02, …, 59	$M	This variable is replaced with the current month in English, possible values: January, February, …, December
$n	This variable is replaced with the current month in number with leading zero, possible values: 1, 2, …, 12	$N	This variable is replaced with the configured SIP Display Name or account name

$o	This variable is replaced with the current month in number with leading zero, possible values: 01, 02, …, 12	$O	-
$p	-	$P	This variable is replaced with the current AM/PM status in upper case, possible values: AM, PM
$r	This variable is replaced with the volume level	$R	-
$s	This variable is replaced with the current second of minute with leading zero, possible values: 01, 02, …, 59	$S	This variable is replaced with the state name of the weather information
$t	-	$T	This variable is replaced with the current hour:minute (am/pm) of the day, in which ":" will flash per second. Depending on user's configuration, it will be displayed as 12 hour or 24 hour format. Possible values: 1:00pm, 13:00
$v	This variable is replaced with 5V power usage alert message when incorrect power is used	$V	This variable is replaced with the configured Account SIP Server host
$w	This variable is replaced with the temperature of the weather information	$W	This variable is replaced with the current day of week and has the following possible values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
$x	This variable is replaced with the humidity of the weather information	$X	This variable is replaced with the configured Account SIP User ID
$y	This variable is replaced with the current year in 2-digit number, for example: 06, 07	$Y	This variable is replaced with the current year in 4-digit number, for example: 2006, 2007 …
$xU	This variable is replaced with account x's configured SIP User ID where x is 0 to 5 for account 1 to account 6. For example, $0U is account 1's SIP User ID if configured.		
$xN	This variable is replaced with account x' s Display Name if filled in, otherwise account x' s Account Name will be used where x is 0 to 5 for account 1 to account 6. For example, $0N is account 1's Display Name if configured. If account 1's Display Name is not configured, $0N will be replaced with account 1's Account Name.		
$xS	This variable is replaced with account x' s SIP Server where x is 0 to 5 for account 1 to account 6. For example, $0S is account 1's SIP Server if configured.		

Note:

To display "$", please use "$$" escape sequence.