Grandstream Networks, Inc.

GXV3140 IP Multimedia Phone

XML Based GUI Configuration Guide

TABLE OF CONTENTS

TABLE OF FIGURES

1 OVERVIEW

The GXV3140 IP Multimedia Phone allows users to customize the GUI desktop layout as well as GUI configurations on the phone, offering users with flexibility and control. The XML GUI configuration includes display/hide certain applications, configure parameters on the phone with specific configuration files, control the display appearance and enable/disable some applications and much more.

This application guide gives a detailed description on how to configure the GUI on the phone using the XML document. The first part will describe the 1) Structural implementation and the hierarchy of the XML document, 2) XML syntax and example XML file for the main control file, 3) XML syntax and an example XML file for configuring the config-file elements and 4) Provide a guide on generating the configuration file for the GXV3140 using the Grandstream Config Tool and downloading it to the phone.

2 STRUCTURAL IMPLEMENTATION

All the GUI configurations on the GXV3140 are implemented and controlled by the XML file, which controls the hierarchy, display and implementation on the phone.

- Control File (menu.xml): GUI configuration document that is used to modify/control the GUI interface on the GXV31XX. It is able to implement the following functions:
  - Modify the display hierarchy of the menu
  - Disable/Enable (Hide/Lock) specific application features.
  - Modify the display of applications on GUI. This includes modifying the display icon/text or display the application with default settings.
  - Specify configuration files and configure the settings for particular applications. These configurations includes :
    - Parameter configurations
    - Function configuration: Display/Hide/Lock specific functions
    - Advanced configuration: Specify configuration files and other resource files so that the advanced settings on the phone can be configured using this information.
  - Other Configurations (reserved for future modifications)

Refer to the diagram below for the basic control flow.

Note: Some of the configurations in the basic control flow diagram below are for demo purposes only. It may change somewhat during real-life implementations.

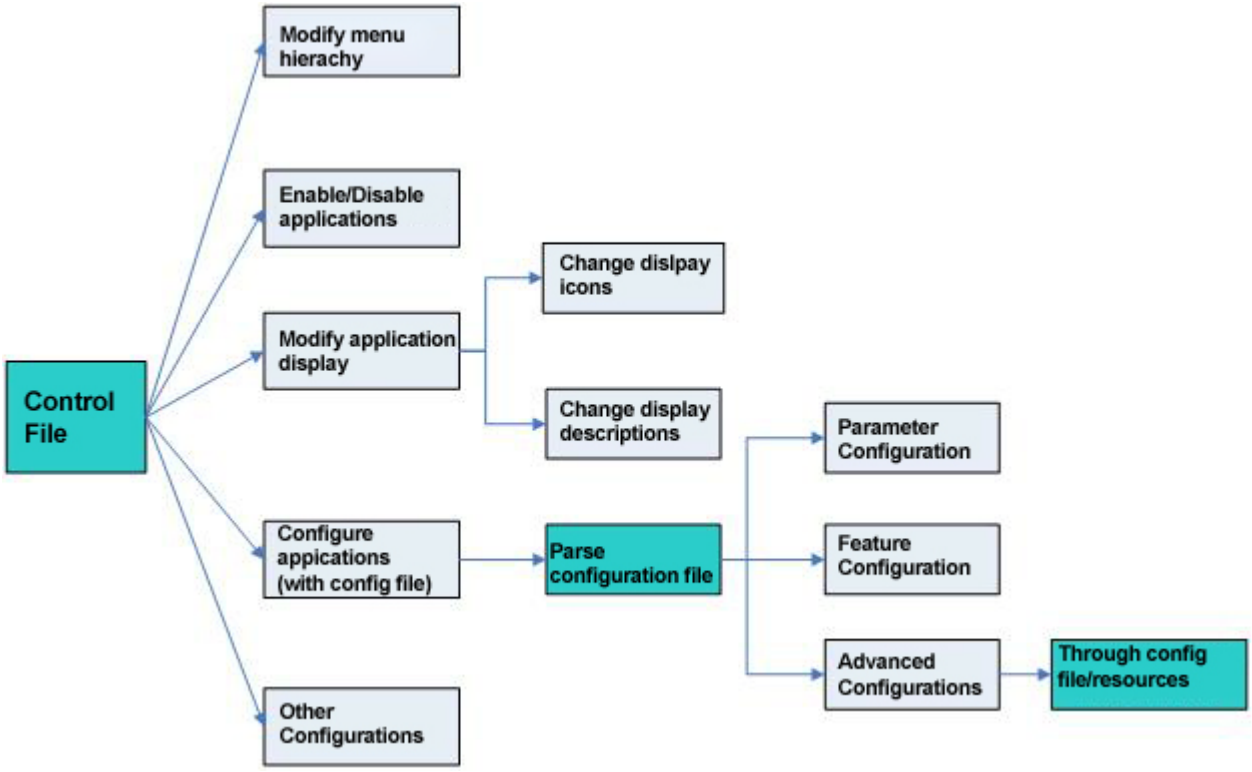


Figure 1: Configuration File Implementation Flow

2.1 ELEMENT DESCRIPTION

The control file supports these following elements:

- **<menu>** element: Root element of the configuration file.
- **<submenu>** element: sub-element used to implement the categorization and the hierarchy of the GUI menu. It can have **<submenu>**, **<item>** or **<link>** as its sub-element**.**
- **<item>** element: Application element that corresponds to a specific application on the phone. It could have a separate configuration file to configure the parameters and features for that particular application.
- **<link>** element: URL element used to implement the link to specific web URL. The GXV31XX will link to and display the URL using the built-in web browser.
- **<config-file> element:** This element specifies the configuration file used for the application, which will apply to the application in the **<item>** element. It can only be used as a sub-element for **<item>** element. Currently, it will work with these 4 applications: Account, Maintenance, Screen Layout, and Screen Display. For more detailed description, please refer to the sections below. (This element is optional)
- **<link-url>** element: This element specifies the URL address of the website to link to. It can only be used as a sub-element for <link> element.

Shared attributes for elements <**submenu**>, <**item**> and <**link**>**:**

- **Icon:** The menu displays static icons when it is not selected. This setting is configuration but optional. If nothing is specified, the phone will use the default menu icon. (This element is optional)

- **dynamic-icon**: The menu displays dynamic icons when it is selected. This setting is configuration but optional. If nothing is specified, the phone will use the default menu icon. (This element is optional)
- **display-name**: The display name for the icons in the menu. This setting is configuration but optional. If nothing is specified, the phone will use the default menu text. (This element is optional) Note: If this parameter is specified, it will overwrite the default menu text.
- **hide**: Show/hide the menu items. 1- Hide/Disable the menu items, 0- Show/Enable the menu items. By default, it is set to 1. (This element is optional)

Unique attributes for elements <submenu> and <item> :

func-name: Menu item ID that is used to identify a specific menu item uniquely. By default, there are some pre-defined settings for the default menu items; this includes dynamic-icon, icon and display-name. For the item element, these are required attributes. For submenu element, these are optional attributes.

Note:

1. By default, we have some pre-defined menu items with func-name. These menu items has default static icon, dynamic icon and display text. The display text will be changed when the phone loads the language file based on the language setting on the GUI interface. Users can also overwrite the default display-text by configuring the display-name attribute. For pre-defined func-name and the corresponding menu item, please refer to the table below.
2. For all self-defined submenu elements and link elements, it is required to configure the icon, dynamic-icon and display-name attribute so that it can be displayed properly on the phone. If these are not configured properly, it may have incorrect display.
3. config-file element: Currently, it will work with these 4 applications: Account, Maintenance, Screen Layout, and Screen Display.
4. All the configuration files (including the control file and the specific application configuration file) use its current directory as the root directory.

func-name	Menu Item	Element Type	Support for config-file element
phonebook	Phone Book	item	No
callhistory	Call History	item	No
messages	Messages	item	No
callfeature	Call Features	item	No
webbrowser	Web Browser	item	No
status	Info	item	No
calendar	Calendar	item	No
alarmclock	Alarm Clock	item	No
calculator	Calculator	item	No
weather	Weather	item	No
rssnews	RSS News	item	No
ipcall	Direct IP Call	item	No
gstris	Gstris	item	No
softkey_def	Softkey	item	No
display	Display	item	No
time	Time	item	No
connection	Network	item	No
gphoto	Photo Viewer	item	No
webalbum	Photo Album	item	No
language	Language	item	No
IM	IM	item	No
filemanager	File Manager	item	No
player	Media Player	item	No

internetradio	Internet Radio	item	No
stock	Stock	item	No
currencies	Currencies	item	No
myprofile	My Profile	item	No
onlinemusic	Online Music	item	No
camera	Camera	item	No
tones	Tones	item	No
photobucket	Photobucket Application	item	No
phanfare	Phanfare Application	item	No
onlinevideo	Online Video	item	No
fxo	FXO Settings	item	No
horoscope	Horoscope	item	No
panoramio	World Photos	item	No
slide_show	Slide Show	item	No
video	Video Settings	item	No
ip2location	IP2Location	item	No
twitter	Twitter Application	item	No
facebook	Facebook Application	item	No
today	Today Application	item	No
account	Accounts	item	Yes
maintenance	Maintenance	item	Yes
screen_layout	Screen Layout	item	Yes
screen_display	Screen Saver	item	Yes
application	Applications	submenu	No
multimedia	Multimedia	submenu	No
settings	Settings	submenu	No
personalize	Personalize	submenu	No
socialnetwork	Social Networks	submenu	No

3 EXAMPLE CONTROL FILE (MENU.XML)

See the demo file (menu.xml) below:

For more details or examples, please download the XML example file package from our website.

```
<?xml version="1.0" encoding="UTF-8"?>
<menu>
   <submenu icon="icon/2.png" dynamic-icon="icon/3.png" display-name="Communication">
      <item func-name="phonebook"/>
      <item func-name="callhistory"/>
      <item icon="icon/slideshow.png" dynamic-icon="icon/slideshow.gif" func-name="messages"/>
   </submenu>
   <submenu func-name="socialnetwork" hide="1">
      <item func-name="IM"/>
      <item func-name="twitter"/>
      <item func-name="facebook"/>
   </submenu>
   <submenu icon="icon/webbrowser.png" dynamic-icon="icon/webbrowser.gif" display-name="Internet">
      <item func-name="webbrowser"/>
      <link icon="icon/Grandstream.png" dynamic-icon="icon/Grandstream.gif" display-
name="Grandstream Network">
         <link-url>http://www.grandstream.com</link-url>
```

```
    </link>
  </submenu>
  <submenu func-name="multimedia" display-name="Media APP">
    <item func-name="internetradio" display-name="My Radio"/>
    <item func-name="player"/>
    <item func-name="gphoto"/>
    <item func-name="webalbum"/>
    <item func-name="gstris"/>
    <item func-name="onlinemusic" hide="1"/>
    <item func-name="onlinevideo"/>
    <item func-name="panoramio"/>
    <item func-name="slide_show"/>
  </submenu>
  <submenu func-name="application">
    <item func-name="calendar"/>
    <item func-name="alarmclock"/>
    <item func-name="calculator"/>
    <item func-name="ipcall"/>
    <item func-name="filemanager"/>
    <item func-name="stock"/>
    <item func-name="currencies"/>
    <item func-name="ip2location"/>
    <item func-name="today"/>
  </submenu>
  <submenu func-name="settings">
    <item func-name="account">
      <config-file>acct.xml</config-file>
    </item>
    <item func-name="connection"/>
    <item func-name="time"/>
    <item func-name="display"/>
    <item func-name="maintenance">
      <config-file>maintenance.xml</config-file>
    </item>
    <item func-name="camera"/>
    <item func-name="callfeature"/>
    <item func-name="video"/>
    <item func-name="fxo"/>
  </submenu>
  <submenu func-name="personalize">
    <item func-name="screen_layout" hide="1">
      <config-file>screen_layout.xml</config-file>
    </item>
    <item func-name="softkey_def"/>
    <item func-name="screen_display">
      <config-file>screen_display.xml</config-file>
    </item>
    <item func-name="language"/>
    <item func-name="myprofile"/>
    <item func-name="tones"/>
    <item func-name="weather"/>
    <item func-name="rssnews"/>
    <item func-name="horoscope"/>
  </submenu>
  <item func-name="status"/>
</menu>
```

Refer to the text below for detailed explanation on the demo file:

1.

```
<submenu icon="icon/2.png" dynamic-icon="icon/3.png" display-name="Communication">
   <item func-name="phonebook"/>
   <item func-name="callhistory" hide="1"/>
   <item icon="icon/slideshow.png" dynamic-icon="icon/slideshow.gif" func-name="messages"/>
</submenu>
```

A sub-menu with the display text “Communication“ and the icons “icon/2.png” and “icon/3.png” is defined here. This sub-menu includes three items/applications: Phonebook, Call History and Message.

2.

```
<submenu icon="icon/2.png" dynamic-icon="icon/3.png" display-name="Communication">
   <item func-name="phonebook"/>
   <item func-name="callhistory" hide="1"/>
   <item icon="icon/slideshow.png" dynamic-icon="icon/slideshow.gif" func-name="messages"/>
</submenu>
```

For the Message application, the default icon is replaced with self-defined icons (slide-show.png and slideshow.gif)

3.

```
<item func-name="onlinemusic" hide="1"/>
```

Disable (hide) the Online Music application.

4.

```
<submenu icon="icon/webbrowser.png" dynamic-icon="icon/webbrowser.gif" display-name="Internet">
   <item func-name="webbrowser"/>
   <link icon="icon/Grandstream.png" dynamic-icon="icon/Grandstream.gif" display-name="Grandstream
Network">
      <link-url>http://www.grandstream.com</link-url>
   </link>
</submenu>
```

Define an icon in GUI menu that links to Grandstream Website.

5.

```
<submenu func-name="settings">
   <item func-name="account">
      <config-file>acct.xml</config-file>
   </item>
   <item func-name="connection"/>
   <item func-name="time"/>
   <item func-name="display"/>
   <item func-name="maintenance">
      <config-file>maintenance.xml</config-file>
   </item>
   <item func-name="camera"/>
   <item func-name="callfeature"/>
   <item func-name="video"/>
   <item func-name="fxo"/>
</submenu>
```

The application/item Account specifies the configuration file acct.xml and the application/item Maintenance specifies the configuration file maintenance.xml. Refer to the sections below for more details on how these XML file works.

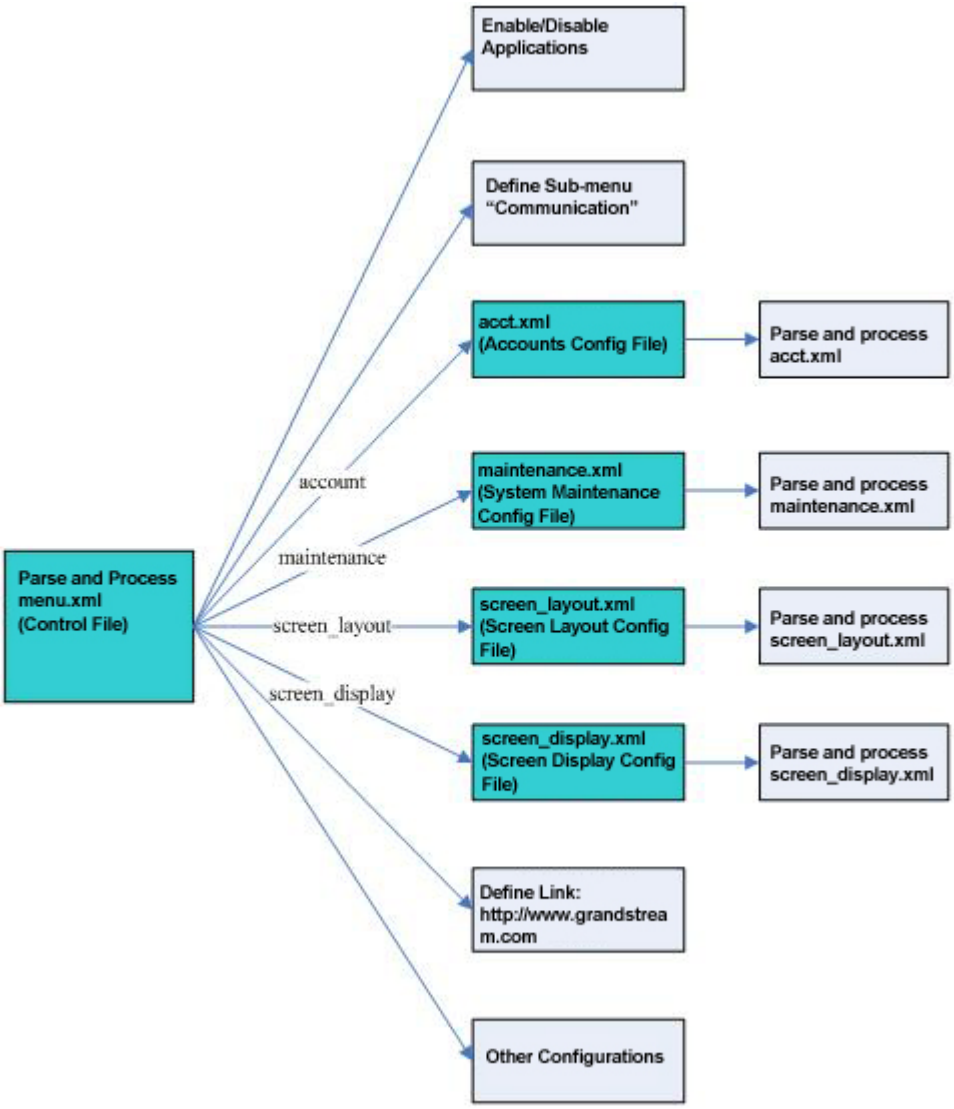


Figure 2: Flow diagram of example demo file

Note: All the configuration files are parsed through the XML parser, therefore it is essential to follow the standard XML schema format and refer to the example configuration files at the end of this document for more descriptions and details. Since the validity of the configuration files are checked with XML Schema, please make sure that all elements and attributes follow the standards to avoid any parsing failure/error.

4 XML CONFIG-FILE ELEMENTS

Users may configure other configurations on the phone using the XML configuration file specified in the main control file menu.xml.

4.1 ACCOUNT CONFIGURATION (ACCT.XML)

```
<item func-name="account">
   <config-file>acct.xml</config-file>
</item>
```

This allows users to configure Accounts using the specified configuration file (e.g. acct.xml). This includes configuring the following: Account SIP server, SIP user ID, Authenticate ID, Authenticate password etc… It also defines settings such as whether the account is active, hiding the account or reordering the accounts.

Refer to the account configuration file (acc.xml) below:

```
<acct-config>
   <account lock="1" hide="0">
      <index>1</index>
      <active>1</active>
      <pos>1</pos>
      <name>Account 2</name>
      <server>192.168.1.20</server>
      <outbound-proxy></outbound-proxy>
      <auth-id>9090</auth-id>
      <auth-pwd>9090</auth-pwd>
      <user-id>9090</user-id>
      <display-name>9090</display-name>
   </account>
   <account lock="1" hide="0">
      <index>2</index>
      <active>1</active>
      <pos>2</pos>
      <name>Account 1</name>
      <server>192.168.1.20</server>
      <outbound-proxy></outbound-proxy>
      <auth-id>8080</auth-id>
      <auth-pwd>8080</auth-pwd>
      <user-id>8080</user-id>
      <display-name>8080</display-name>
   </account>
   <account lock="1" hide="0">
      <index>3</index>
      <active>1</active>
      <pos>3</pos>
      <name>Account 3</name>
      <server>192.168.1.20</server>
      <outbound-proxy></outbound-proxy>
      <auth-id>7070</auth-id>
      <auth-pwd>7070</auth-pwd>
      <user-id>7070</user-id>
      <display-name>7070</display-name>
   </account>
</acct-config>
```

The various elements and attributes are described below:

- <acct-config>: Account configuration element
- <account>: Account element, has the following attributes: lock and hide.
  - lock: Valid values are 0 or 1. 1- Account settings are locked and cannot be changed by the user, 0- Account settings are not locked and can be changed by the user.
  - hide: Valid values are 0 or 1. Currently, this is the same as the lock attribute.

- <index>: Actual account index, where the accounts has not been re-ordered. Valid values are from 1~3.

- <active>: Specifies whether the configuration is active or not. Valid values are 0 or 1. 1- Active, 0- Inactive.
- <pos>: Account position index. Allows the reordering of accounts, such as moving Account 1 to Account 3. Valid values are from 1~3.
- <name>: Account Name. ((Type- String)
- <server>: SIP server URL. (Type- String)
- <outbound-proxy>: Outbound proxy server URL. (Type- String)
- <auth_id>: SIP Account Authenticate ID. (Type- String)
- <auth_pwd>: SIP Account Authenticate Password. (Type- String)
- <user-id>: SIP User ID. (Type- String)
- <display-name>: Display Name. (Type- String)

4.2 MAINTENANCE SETTINGS CONFIGURATION (MAINTENANCE.XML)

```
<item func-name="maintenance">
   <config-file>maintenance.xml</config-file>
</item>
```

Maintenance Settings configurations is used to configure settings related to firmware upgrade and configuration file upgrade.

Refer to the maintenance settings configuration file (maintenance.xml) below:

```
<maintenance>
   <factory-reset lock="0" hide="1"/>
   <firmware-config lock="1" hide= "0"/>
</maintenance>
```

The various elements and attributes are described below:

- <maintenance>，Configuration file root element
  - <factory-reset>: Restore to factory default settings, has the following attributes:
    - lock : Valid values are 0 or 1. 1- Setting for factory reset is locked (Users can see it but cannot operate), 0- Setting for factory reset is not locked (Users can operate)
      hide : Valid values are 0 or 1.，1- Setting for factory reset is hidden , 0- Setting for factory reset is shown.

    Note: The hide attribute has higher priority than lock attribute.

- <firmware-config>: Configurations for upgrade mode, firmware server path, config server path. It has the following attributes:
    - lock : Valid values are 0 or 1. 1- Configuration settings are locked and cannot be changed by the user, 0- Configuration settings are not locked and can be changed by the user.
      hide : Valid values are 0 or 1. 1- Hide the configurations, 0-Show the configurations.

    Note: The hide attribute has higher priority than lock attribute.

4.3 SCREEN LAYOUT CONFIGURATION (SCREEN_LAYOUT.XML)

```
<item func-name="screen_layout" hide="1">
   <config-file>screen_layout.xml</config-file>
</item>
```

The Screen Layout Configuration is used to control the desktop screen layout, enable/disable certain desktop applications and determine whether or not to use custom layout etc…

Refer to the screen layout configuration file (screen_layout.xml) below:

```
<?xml version="1.0" encoding="UTF-8"?>
<screen-layout>
   <version>2</version>
```

```
  <layout-type>1</layout-type>
  <small-layout>
    <desktop1>
      <item position="left_top">RSS News</item>
      <item position="right_top">Digital Clock</item>
      <item position="right_bottom">Information</item>
    </desktop1>
    <desktop2>
      <item position="left_top">Weather</item>
      <item position="right_top">Calendar</item>
    </desktop2>
    <desktop3/>
    <desktop4/>
  </small-layout>
  <large-layout>
    <desktop1>
      <item position="left">Information</item>
      <item position="right">Analog Clock</item>
    </desktop1>
    <desktop2>
      <item position="left">RSS News</item>
      <item position="right">Stock</item>
    </desktop2>
    <desktop3>
      <item position="left">Calendar</item>
      <item position="right">Weather</item>
    </desktop3>
    <desktop4>
      <item position="left">Horoscope</item>
      <item position="right">Today</item>
    </desktop4>
  </large-layout>
  <custom-layout>
    <file>gxv31xx_layout.xml</file>
  </custom-layout>
</screen-layout>
```

The various elements and attributes are described below:

- <screen-layout>: Configuration file root element
- <version>: configuration file version. This will increment if there are changes in the configuration file element, to identify the different configuration file version.
- <layout-type>: Valid values are the following:
  - 0: Small Layout. Configure the two desktops and the desktop applications shown according to the configuration specified in <small-layout>
  - 1: Large Layout. Configure the four desktops and the desktop applications shown according to the configuration specified in <large-layout
  - 2: Customized Layout. Use the custom layout configuration file specified in <custom-layout> to configure the custom layout for desktop.
- <small-layout>: Small-layout configuration element
  - <desktopX>: Configure the layout for Desktop X, where X is a value from 1 ~ 2. Every desktop has four windows (top left window, top right window, bottom left window, bottom right window) for the display of desktop applications.
    - <item>: Specifies the application on the desktop (Type: String). Valid values are the following:
      - RSS News: RSS News desktop application

- Weather: Weather desktop application
- Analog Clock: Analog Clock desktop application
- Digital Clock: Digital Clock desktop application
- Information: System Information desktop application
- Calendar: Calendar desktop application
- Stock: Stock desktop application
- Horoscope: Horoscope desktop application
- Today: Today desktop application

◆ <item> element has "position" attributes (String type), valid values are:
- left_top: top left part of the window
- right_top: top right part the window
- left_bottom: bottom left part of the window
- right_bottom: bottom right part of the window

- <large-layout>: Large-layout configuration element
  - <desktopX> Configure the layout for Desktop X, where X is a value from 1 ~ 4. Every desktop has two windows (left window, right window) for the display of desktop applications.
    - ◆ <item>: Desktop application element (Type: String). The valid values are the same as the <item> element in <small-layout>
    - ◆ <item> element has "position" attributes (Type: String), valid values are:
      - left : left window
      - right : right window
- <custom-layout> Custom layout element
  - <file>: Specifies the custom layout configuration file name. The phone will parse the custom layout configuration file and configure the screen layout accordingly. All the required resources should be located in the same directory as the custom layout configuration file to avoid any parsing error. For more details and the syntax for configuring the XML based customizable screen, please refer to Appendix: GXV31XX XML Based Customizable Screen.

4.4 SCREEN DISPLAY CONFIGURATION (SCREEN_DISPLAY.XML)

```
<item func-name="screen_display" hide="1">
   <config-file>screen_display.xml</config-file>
</item>
```

This is used to control the configurations related to screen display, for example, screensaver, screensaver refresh time, and background picture etc…

Refer to the screen display configuration file (screen_display.xml) below:

```
<?xml version="1.0" encoding="UTF-8"?>
<screen-display>
   <version>1</version>
   <screensaver>
      <type>0</type>
      <time>5</time>
      <lcdoff>15</lcdoff>
      <folder>icon</folder>
      <interval>10</interval>
      <http/>
   </screensaver>
   <background>
      <type>0</type>
      <picture>4.png</picture>
      <image-pos>0</image-pos>
   </background>
</screen-display>
```

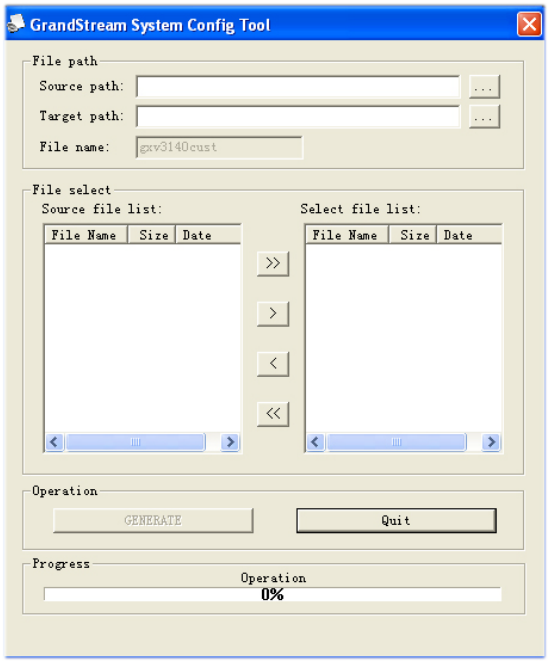
The various elements and attributes are described below:

- <screen-display>: Configuration file root element
- <version>: configuration file version. This will increment if there are changes in the configuration file element, to identify the different configuration file version.
- <screensaver>
  - <type>: screensaver type, valid values are:
    - 0: Default Folder
    - 1: Customized Folder
    - 2: HTTP URL
    - 3: World Photos
    - 4: Flickr
    - 5: Photobucket
    - 6: Phanfare
  - <time>: Screensaver timeout period (in minutes). Valid values are integers from 0~9999, where 0 is used to turn off the screensaver functions. The default value is 5.
  - <lcdoff>: LCD Auto Power Off Time (in minutes). Valid values are integers from 0-9999, where 0 is used to turn this function off. The default value is 15.
  - <folder>: The path to the customized folder that includes the picture files used for the screensaver. The configuration file should be placed under the same folder as the picture, so that the phone is able to locate and parse the file.
  - <interval>: Screensaver picture refresh interval (in sec). Valid values are integers from 10-999. The default value is 10.
  - <http>: HTTP photo URL
- <background>: background configuration element
  - <type>: Desktop background type. Valid values are the following:
    - 0: Default Background
    - 1: Pattern 1
    - 2: Pattern 2
    - 3: Pattern 3
    - 4: Customized picture
  - <picture>: Contains the custom image file name in relative path.
  - <image-pos>: Specifies the image position. Valid values are the following:
    - 0: Stretch
    - 1: Whole Tile
    - 2: Scale Tile

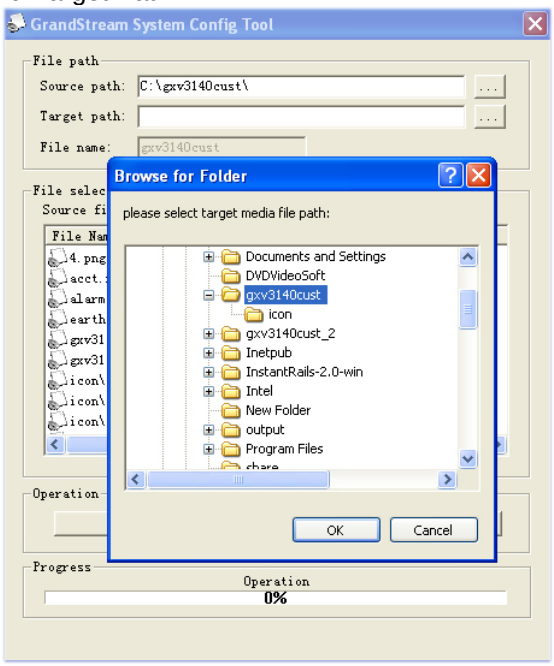
5 CONFIGURE THE GUI CUSTOMIZATION ON THE GXV3140

Grandstream has developed a Configuration Generator Tool for the GUI customization purposes on the GXV3140. This Configuration Generator Tool can be downloaded at: GXV3140 GUI Customization Tool

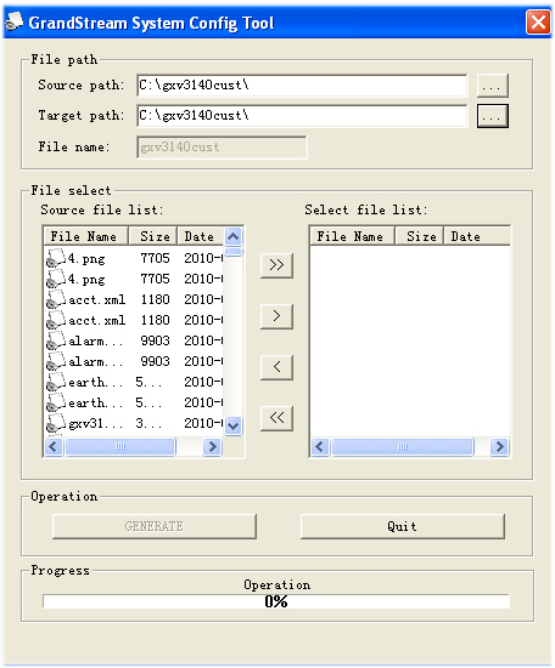
Step 1: Launch the Grandstream System Config Tool on your PC by executing the ConfigTool.exe file.



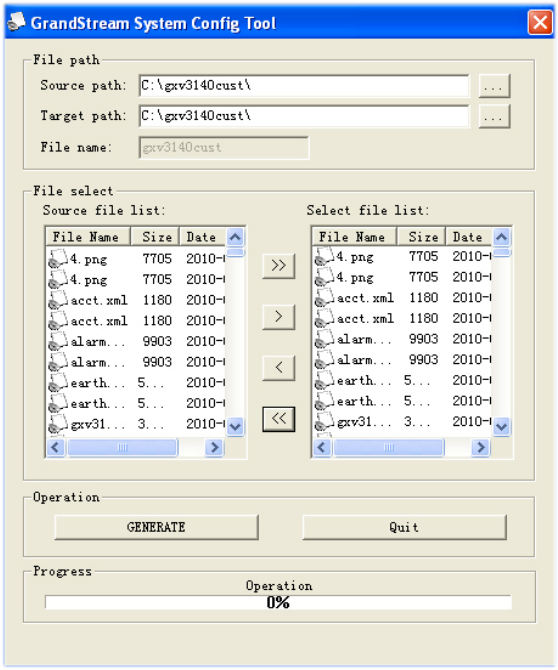
Step 2: Specify the Source and Target Path.



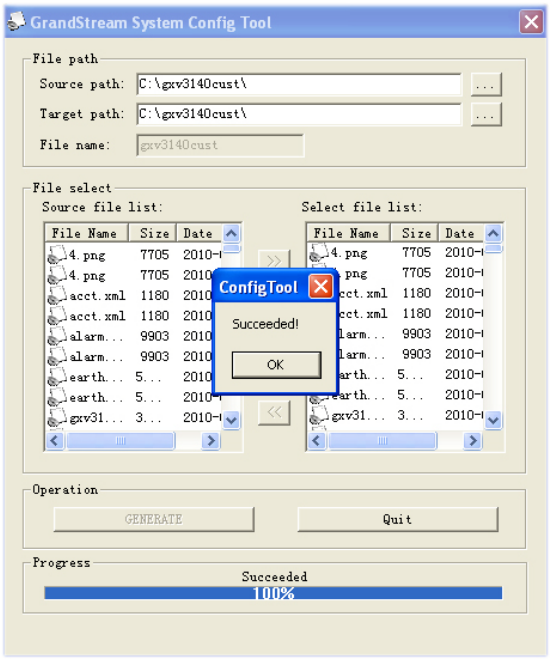
Step 3: Move the source files over to the “Select File List



Step4: Press the “Generate Button”



The System Config Tool will start parsing the files and will generate the **gxv3140cust** file when it has succeeded. This file should be available in the target path specified.



Place the **gxv3140cust** file in the local TFTP/HTTP server root directory and change the firmware server path on the GXV3140 to point to this server. Reboot the phone. By default, the GXV3140 will send out the request to download the gxv3140cust file every time at phone boot-up. Once the GXV3140 has successfully downloaded and parsed the file, the settings will take place and the user will see the customized GUI menus.

6 REFERENCES

[1] Appendix A: GXV31XX XML Based Customizable Screen
[2] Config Tool and Sample Files: GXV3140 GUI Customization Tools

7 APPENDIX: GXV31XX XML BASED CUSTOMIZABLE SCREEN

This part of the document describes how to make a custom screen on the GXV31xx with the XML configuration file. The Gxv31xx serial products support three types of layouts: small layout, large layout and custom layout. You can select the custom layout mode and select the XML configure file which includes the layout information, save to make it take effect.

7.1 EXAMPLE XML FILE FOR CUSTOMIZED SCREEN LAYOUT

The XML configuration file includes the format which decide how and where to display the screen items. At present, there are four screen item types: the panel (only one), pictures, strings and functions.

The following is an example XML configuration file:

```
<Screen>
  <IdleScreen>
    <DesktopLayout>large</DesktopLayout>
    <Desktop id="1">
      <DisplayString size="16" color="#ecf164" style="bold" halignment="0.5" valignment="0.5">
        <StringValue> Call Me </StringValue>
        <X>10</X>
        <Y>10</Y>
        <Width>200</Width>
        <Height>-1</Height>
      </DisplayString>
      <DisplayPicture scaled="true" preserve_ratio="true">
        <FileName>music.png</FileName>
        <X>0</X>
        <Y>0</Y>
        <Width>50</Width>
        <Height>50</Height>
      </DisplayPicture>
      <DisplayPicture>
        <FileName>earth.gif</FileName>
        <X>240</X>
        <Y>20</Y>
      </DisplayPicture>
      <DisplayString size="13" color="#ffedcd" style="italic" valignment="1.0">
        <RegisterShow account="acct_1">false</RegisterShow>
        <StringValue>Account 1 Unregister</StringValue>
        <X>240</X>
        <Y>150</Y>
      </DisplayString>
      <DisplayString size="13" color="#52e3da" style="bold" halignment="1.0" valignment="0.5">
        <StringValue>$Y-$o-$D $W</StringValue>
        <X>10</X>
        <Y>140</Y>
        <Width>200</Width>
        <Height>-1</Height>
      </DisplayString>

      <Function>
        <FuncRssNews layout="large">
          <DesktopID>2</DesktopID>
```

```
            <X>0</X>
            <Y>10</Y>
          </FuncRssNews>
          <FuncWeather layout="large">
            <DesktopID>2</DesktopID>
            <X>240</X>
            <Y>10</Y>
          </FuncWeather>
          <FuncCalendar layout="large">
            <DesktopID>3</DesktopID>
            <X>10</X>
            <Y>10</Y>
          </FuncCalendar>
        </Function>

        <Panel show="true"/>

    </IdleScreen>
    <CallScreen>
    </CallScreen>

</Screen>
```

7.2 XML SYNTAX EXPLANATION

7.2.1 Root Element "Screen"

The XML document has root element called Screen; it contains sub-elements called IdleScreen and CallScreen. At present, we require the IdleScreen element to be present and the Call Screen element is optional.

```
<xsd:element name="Screen">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="IdleScreen">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="DesktopLayout" type="LayoutType" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="Desktop" minOccurs="1" maxOccurs="3"/>
            <xsd:element ref="Function" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="Panel" minOccurs="0" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="CallScreen" minOccurs="0" maxOccurs="1">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element
```

7.2.2 Element "IdleScreen"

This element defines four components that are makes up the idle screen. These components are defined as elements.

```
<xsd:element name="Screen">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="IdleScreen">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="DesktopLayout" type="LayoutType" minOccurs="1" maxOccurs="1"/>
            <xsd:element ref="Desktop" minOccurs="1" maxOccurs="3"/>
            <xsd:element ref="Function" minOccurs="0" maxOccurs="1"/>
            <xsd:element ref="Panel" minOccurs="0" maxOccurs="1"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="CallScreen" minOccurs="0" maxOccurs="1">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element
```

7.2.2.1 Element "DesktopLayout"

This element defines the whole layout of the screen; its value can only be small or large. If it is "small", it defines two desktops on the screen; if it is "large", it defines three desktops on the screen. You can put different things on different desktops and switch between them.

```
<xsd:simpleType name="LayoutType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="small"/>
    <xsd:enumeration value="large"/>
  </xsd:restriction>
</xsd:simpleType>
```

7.2.2.2 Element "Desktop"

This element defines the display items on one of the desktops, and they are displayed in the order they appear in the XML and later objects may overwrite/corrupt previous object.

You can put two types of objects: the picture and the string. Actually, there is a third object: the function, which will be described on the following chapter. It has an attribute "id", its value presents which desktop you want to configure.

```
<xsd:attribute name="id" type="DesktopIDTYpe" use="required"/>
<xsd:simpleType name="DesktopIDTYpe">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="1"/>
    <xsd:maxInclusive value="4"/>
  </xsd:restriction>
</xsd:simpleType>
```

7.2.2.2.1 Element "DisplayPicture"

The "DisplayPicture" element puts a picture at the special position and with the special style. It contains the following sub-elements and attributes:

```
<xsd:element name="DisplayPicture" minOccurs="0" maxOccurs="unbounded" nillable="true">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="RegisterShow" minOccurs="0" maxOccurs="1">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:boolean">
              <xsd:attribute name="account" type="AccountType" use="required"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="FileName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="X" type="xsd:integer" minOccurs="1" maxOccurs="1" default="0"/>
      <xsd:element name="Y" type="xsd:integer" minOccurs="1" maxOccurs="1" default="0"/>
      <xsd:element name="Width" type="xsd:integer" minOccurs="0" maxOccurs="1" default="-1"/>
      <xsd:element name="Height" type="xsd:integer" minOccurs="0" maxOccurs="1" default="-1"/>
    </xsd:sequence>
    <xsd:attribute name="scaled" type="xsd:boolean" use="optional" default="false"/>
    <xsd:attribute name="preserve_ratio" type="xsd:boolean" use="optional" default="false"/>
  </xsd:complexType>
</xsd:element>
```

Element "RegisterShow" :

This element is optional, and it used to decide when to show the picture. The element present or absent with the value means:

- when present and the value is "**true**" – show when registered
- when present and the value is "**false**" - show when unregistered
- when absent - show always regardless of the register status

And it has an attribute called "**account**" to make the picture relate to the special account:

- "acct_1" - relate to account 1
- "acct_2" - relate to account 2
- "acct_3" - relate to account 3
- "acct_all" -relate to all of the accounts

```
<xsd:element name="RegisterShow" minOccurs="0" maxOccurs="1">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:boolean">
        <xsd:attribute name="account" type="AccountType" use="required"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

<xsd:simpleType name="AccountType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="acct_1"/>
    <xsd:enumeration value="acct_2"/>
    <xsd:enumeration value="acct_3"/>
    <xsd:enumeration value="acct_all"/>
  </xsd:restriction>
</xsd:simpleType>
```

Element "FileName":
This specifies the name of the picture that will be displayed on the desktop. It is mandatory.

Element "X":
Decide the x coordinate where the picture puts on the desktop. It is mandatory.

Element "Y":
Decide the y coordinate where the picture puts on the desktop. It is mandatory.

Element "Width":
Decide the width of the picture, this will only take effects when the attribute "scaled" set to "true", otherwise the original width of the picture will be kept. This element is optional. When it is absent or when the value is equal to -1, the width will be kept to the original width of the picture.

Element "Height":
Decide the height of the picture, only take effects when the attribute "scaled" set to "true", otherwise keep the original height of the picture. This element is optional. When it is absent or when the value is equal to -1, the height is kept the original height of the picture.

Attribute "scaled":
Whether to scale the picture to the size Width * Height .The value must be "true" or "false". This attribute is optional. When it is absent or when the value is equal to "false", the picture will not be scaled.

Attribute "preserve_ratio":
Decide whether to preserve the aspect ratio of the picture when scaling. Only take effects when the attribute "scaled" set to "true". The value must be "true" or "false". This attribute is optional. When it is absent or when the value is equal to "false", the ratio of the picture is not preserved.

7.2.2.2.2 Element "DisplayString"

This element carries the information on how a string is to be rendered on screen. It has the following elements and attributes:

Element "RegisterShow":
The same function as the "RegisterShow" element in "DisplayPicture" element. Shows the string when account is registered, unregistered or always. It is optional.

Element "StringValue":
This specifies the string which will be displayed on the screen. It is mandatory.

Element "X":
The same as the "DisplayPicture", the x coordinate where the string is placed. It is mandatory.

Element "Y":
The same as the "DisplayPicture", the y coordinate where the string is placed. It is mandatory.

Element "Width":
The total width of the string placed on the screen. It can be used to make horizontal alignment. It is optional. When it is absent or when value is -1, the width is decided by the system.

Element "Height":
The total height of the string when put to the screen. It can be used to make the vertical alignment. It is optional. When it is absent or when value is -1, the height is decided by the system.

Attribute "size":

This specifies the font size of the string. The value must be between 5 and 16. This attribute is optional. The system will use the default system font size color if it is not specified.

```
<xsd:simpleType name="FontSizeType">
  <xsd:restriction base="xsd:integer">
    <xsd:minInclusive value="5"/>
    <xsd:maxInclusive value="16"/>
  </xsd:restriction>
</xsd:simpleType>
```

Attribute "color":

This specifies the font color of the string. Possible value is the hexadecimal RGB value in the form #RRGGBB.

This attribute is optional. The system will use the default system color if it is not specified.

Attribute "style":

This specifies the font style of the string (bold or italic). This attribute is optional. The system will use the default setting if it is not specified.

```
<xsd:simpleType name="FontStyleType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="bold"/>
    <xsd:enumeration value="italic"/>
    <xsd:enumeration value="bold italic"/>
  </xsd:restriction>
</xsd:simpleType>
```

Attribute "halignment":

This specifies the horizontal alignment of string, the value ranges from 0 (left) to 1 (right). 0.5 is the center. It is optional, when it is absent, left alignment will be used.

Attribute "valignment":

This specifies the vertical alignment of the string, the value ranges from 0 (top) to 1 (bottom). 0.5 is the center. It is optional, when absent, top alignment will be used.

The string can contain dynamic contents. Currently, we support the following 24 system variables that will be substituted with dynamic contents at run-time.

- 1. $W: This variable is replaced with the current day of week and has the following possible values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
- 2. $N: This variable is replaced with the configured Account 1 Display Name.
- 3. $X: This variable is replaced with the configured Account 1 SIP User ID.
- 4. $V: This variable is replaced with the configured Account 1 SIP Server.
- 5. $A: This variable is replaced with the configured Account 2 Display Name.
- 6. $B: This variable is replaced with the configured Account 2 SIP User ID.
- 7. $C: This variable is replaced with the configured Account 2 SIP Server.
- 8. $E: This variable is replaced with the configured Account 3 Display Name.
- 9. $F: This variable is replaced with the configured Account 3 SIP User ID.
- 10. $G: This variable is replaced with the configured Account 3 SIP Server.
- 11. $I: This variable is replaced with the system IP address.
- 12. $D: This variable is replaced with the current day of month with leading zero, possible values: 01, 02, ..., 31.
- 13. $d: This variable is replaced with the current day of month without leading zero, possible values: 1, 2, ..., 31.

- 14. $M: This variable is replaced with the current month in English, possible values: January, February, ..., December
- 15. $o: This variable is replaced with the current month in number with leading zero, possible values: 01, 02, ..., 12
- 16. $n: This variable is replaced with the current month in number without leading zero, possible values: 1, 2, ..., 12
- 17. $Y: This variable is replaced with the current year in 4-digit number, for example: 2006, 2007 ...
- 18. $y: This variable is replaced with the current year in 2-digit number, for example: 06, 07 ...
- 19. $P: This variable is replaced with the current AM/PM status in upper case, possible values: AM, PM
- 20. $p: This variable is replaced with the current AM/PM status in lower case, possible values: am, pm
- 21. $H: This variable is replaced with the current hour of day in 24-hour representation with leading zero, possible values: 00, 02, ..., 23
- 22. $h: This variable is replaced with the current hour of day in 12-hour representation with leading zero, possible values: 01, 02, ..., 12
- 23. $m: This variable is replaced with the current minute of hour with leading zero, possible values: 01, 02, ..., 59
- 24. $s: This variable is replaced with the current second of minute with leading zero, possible values: 01, 02, ..., 59

Note: If you want to display the "$" sign, you will use "$$" escape sequence.

7.2.2.3 Element "Function"

This element defines whether to display the factory functions on the desktop or not, the user can choose which desktop and where to show the functions. It contains six factory functions and all of them are optional.

FuncRssNews
Create an RSS news program on the screen and it will update periodically.
FuncWeather
Create a weather update program on the screen, and it will update periodically.
FuncCalendar
Create a calendar on the screen, and it will update periodically.
FuncAnalogClock
Create a analog clock on the screen, and it will update periodically.
FuncDigtalClock
Create a digital clock on the screen, and it will update periodically.
FuncInfo
Create a status report on the screen (including the system IP address and SIP information). This will be updated periodically.

Each function has the following elements and attributes:
Element "DesktopID":
The value is the desktop number which the function will be put. It is mandatory.

Element "X":
The same as the "DisplayPicture", the x coordinate where the function is displayed. It is mandatory, and the default value is 0.

Element "Y":
The same as the "DisplayPicture", the y coordinate where the function is displayed. It is mandatory, and the default value is 0.

Attribute "layout":

The function layout, possible value: “small” or “large”.
For the functions: FuncAnalogClock, FuncDigtalClock and FuncInfo. If the “small” is chosen, the function will take a quarter of the screen; If the “large” is chosen, the function will take half of the screen.
For other functions such as FuncRssNews, FuncWeather and FuncCalendar: The function will always take the half of the screen, so the “layout” attribute will have no effects.

7.2.2.4 Element “Panel”

Attribute “show”

The element contains an attribute **“show”** which decides whether to display the panel on the top of the screen or not. It is optional, if the attribute is absent or “true”, the panel will be shown..
The panel includes the networks status icon; the SIP accounts status icons, time/date/desktop status on the right corner. It also has a loading hint on the middle of the panel when loading programs.

7.3 How to download the XML configuration file to the GXV3140

To download the customized screen layout XML file to the GXV3140, select MENU->Personalize-> Screen Layout. In the “Screen Layout” application, select Custom Layout.

Once selected, users have the option of choosing to download from a local source (from the phone folder or through the USB flash drive) or to download from a TFTP/HTTP server.
To download from a TFTP/HTTP server, the user will need to select the download mode and specify the server URL for the download server.

After the user presses the “Download” button, the phone will send out a request to the server to download the configuration file (custom_layout.xml). If the download and the parse of the XML file are successful, the phone will display a message to inform the user. The user can then press “Save” to save the settings. The phone should display the custom layout on the desktop after the settings take place.