Grandstream Networks, Inc.

XML Based Idle Screen Customization Guide

GXP21xx IP Phone/GXP1450 IP Phone

TABLE OF CONTENTS

XML BASED IDLE SCREEN CUSTOMIZATION GUIDE

OVERVIEW

GXP21xx/GXP1450 supports the XML based idle screen customization. The designs of the displayed information and layout depend highly upon personal preferences and requirements and have since been requested by various customers for an API to be able to customize the screen.

This document specifies the Grandstream XML Customizable Screen API design that will be used on GXP-21xx/GXP1450.

HOW IT WORKS AND CONFIGURATION

A new set of configuration options will be introduced as following:

- Enable Idle Screen XML Download (P340): No/YES, HTTP/YES, TFTP (Default No). Possible values: 0 (No)/1 (HTTP)/2 (TFTP), other values ignored.
- Download Screen XML At Boot-up (P1349): No/Yes (Default No). Possible values: 0 (No)/1 (Yes), other values ignored.
- Use Custom filename (P1343): No/Yes (Default No). Possible values: 0 (No)/1 (Yes), other values ignored.
- Idle Screen XML Server Path (P341): This is a string of up to 128 characters that should contain a path to the XML file. It MUST be in the host/path format. For example: "directory.grandstream.com/engineering"

The feature will be activated when "Enable Idle Screen XML Download" is set to YES (HTTP or TFTP) AND a valid "Idle Screen XML Server Path" is set.

This feature does not automatically download the XML file in the path even when activated. Users could set "Download Screen XML At Boot-up" to Yes then it will be downloaded next time when the phone boots up. Or users also have the following 2 options added to the LCD GUI Menu-> Preference:

Download SCR XML
Erase Custom SCR

Users shall select Download SCR XML under Preference Keypad menu to start the download process. The phone will attempt to download the XML file specified in "Idle Screen XML Server Path".

If "Use Custom filename" is set to "No", the XML file name must be gs_screen.xml. The "Idle Screen XML Server Path" is the folder path where gs_screen.xml is located (For example, grandstream.com/directory/custfiles). If "Use Custom filename" is set to "Yes", users could name the file as their preference. In this case, the "Idle Screen XML Server Path" has to be specified to the name of the XML file (For example, grandstream.com/directory/custfiles/gxp2100xmlcust.xml).

Once the XML is successfully downloaded it will be parsed and be effective right away. The file should also be saved for future use.

XML SNYTAX

The XSD file tells the user what the XML file should look like. User does not need to check the structure of the file manually as this process can be automated. One such websites that does this is: http://www.xmlme.com/Validator.aspx

This paragraph will explain how to use the website mentioned above to do the syntax checking.

1. Clear the XML schema and XML document fields on the website.
2. Copy the lines below to the XSD file into the XML schema field.
3. Copy the user idle screen file into the XML document field on the website.
4. Click “validate”.
5. Verify if the validation result is successful. If not, there is structure error in the user idle screen file.

XSD file:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="Screen">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="LeftStatusBar" minOccurs="1" maxOccurs="1" >
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Layout" type="GeneralLayoutTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
       <xs:element name="Account" type="GeneralTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
    <xs:element name="IdleScreen" minOccurs="1" maxOccurs="4">
     <xs:complexType>
      <xs:sequence>

       <xs:element name="ShowStatusLine" type="xs:boolean" minOccurs="0" maxOccurs="1" default="true"/>
       <xs:element name="DisplayBitmap" type="BitmapType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
       <xs:element name="DisplayString" type="StringType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
       <xs:element name="DisplaySet" type="GeneralListHolderType" minOccurs="0" maxOccurs="1"
nillable="true"></xs:element>
       <xs:element name="SoftKeys" minOccurs="0" maxOccurs="1" nillable="true">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="Layout" type="GeneralLayoutTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
          <xs:element name="ButtonShape" type="GeneralTemplateType" minOccurs="1"
maxOccurs="unbounded"></xs:element>
          <xs:element maxOccurs="12" name="SoftKey" type="SoftKeyType">
          </xs:element>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>



 <xs:complexType name="GeneralLayoutTemplateType">
  <xs:sequence>
   <xs:element name="DisplayList" type="ListHolderType" minOccurs="1" maxOccurs="1"></xs:element>
   <xs:element name="DisplayBitmap" type="BitmapType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
   <xs:element name="DisplayString" type="StringType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
  </xs:sequence>
          <xs:attribute name="height" type="xs:int" use="optional" default="0"></xs:attribute>
  <xs:attribute name="width" type="xs:int" use="optional" default="0"></xs:attribute>
  <xs:attribute name="padding-top" type="xs:int" use="optional" default="0"></xs:attribute>
  <xs:attribute name="padding-bottom" type="xs:int" use="optional" default="0"></xs:attribute>
 </xs:complexType>

 <xs:complexType name="GeneralTemplateType">
```

```
  <xs:sequence>
    <xs:element name="DisplayBitmap" type="BitmapType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
    <xs:element name="DisplayString" type="StringType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
  </xs:sequence>
  <xs:attribute name="height" type="xs:integer"/>
  <xs:attribute name="width" type="xs:integer"/>
  <xs:attribute name="id" type="xs:ID" use="optional"></xs:attribute>
 </xs:complexType>

 <xs:complexType name="ListHolderType">
  <xs:sequence>
    <xs:element name="X" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
    <xs:element name="Y" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
  </xs:sequence>
 </xs:complexType>

 <!-- DataList type  -->
 <xs:complexType name="GeneralListHolderType">
  <xs:sequence>
    <xs:element name="ItemTemplate" type="GeneralTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string"/>
  <xs:attribute name="x" type="xs:integer" default="0"/>
  <xs:attribute name="y" type="xs:integer" default="0"/>
  <xs:attribute name="maxNumberOfRecord" type="xs:integer" default="1"/>
  <xs:attribute name="dataSourceName" type="xs:string" use="required"/><!-- [$Stock, $Currency, ($NewsFeed), ($Weather) and
so on] -->
  <xs:attribute name="displayDirection" default="vertical">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="vertical"/>
        <xs:enumeration value="horizontal"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
 </xs:complexType>

 <xs:complexType name="BitmapType">
  <xs:sequence>
    <xs:element name="Bitmap" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="X" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
    <xs:element name="Y" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
    <xs:element name="displayCondition" type="IdleScreenConditionType" minOccurs="0" maxOccurs="1" nillable="true"/>
  </xs:sequence>
  <xs:attribute name="isfile" type="xs:boolean"/>
  <xs:attribute name="isflash" type="xs:boolean"/>
  <xs:attribute name="isrenew" type="xs:boolean"/>
 </xs:complexType>

 <xs:complexType name="StringType">
  <xs:sequence>
    <xs:element name="DisplayStr" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="X" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
    <xs:element name="Y" type="xs:integer" minOccurs="1" maxOccurs="1" default="0"/>
    <xs:element name="displayCondition" type="IdleScreenConditionType" minOccurs="0" maxOccurs="1" nillable="true"/>
  </xs:sequence>

  <xs:attribute name="font" default="regular">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="regular"/>
        <xs:enumeration value="bold"/>
        <xs:enumeration value="unifont"/>
        <xs:enumeration value="time"/>
        <xs:enumeration value="numberfont"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="halign" default="left">
    <xs:simpleType>
```

```
      <xs:restriction base="xs:string">
        <xs:enumeration value="left"/>
        <xs:enumeration value="center"/>
        <xs:enumeration value="right"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="color" default="Black">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Black"/>
        <xs:enumeration value="White"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="isflash" type="xs:boolean"/>
</xs:complexType>

<xs:complexType name="IdleScreenConditionType">
  <xs:sequence>
    <xs:element name="conditionType">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="alwaysDisplay"/>
          <xs:enumeration value="missCall"/>
          <xs:enumeration value="callFwded"/>
          <xs:enumeration value="callFwdCancelled"/>
          <xs:enumeration value="dnd"/>
          <xs:enumeration value="headsetMode"/>
          <xs:enumeration value="signIn"/>
          <xs:enumeration value="signOut"/>
          <xs:enumeration value="backSpace"/>
          <xs:enumeration value="XmlApp"/>
          <xs:enumeration value="SubScreen"/>
          <xs:enumeration value="nfsMountOk"/>
          <xs:enumeration value="nfsMountFailed"/>
          <xs:enumeration value="coredump"/>
          <xs:enumeration value="networkUp"/>
          <xs:enumeration value="hasIM"/>
          <xs:enumeration value="hasVM_IM"/>
          <xs:enumeration value="hasVoiceMail"/>
                              <xs:enumeration value="hasDialedCalllog"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="negate" type="xs:boolean" default="false"/>
</xs:complexType>

<xs:complexType name="EmptyType">
  <xs:sequence />
</xs:complexType>

<xs:complexType name="IdleActionType">
  <xs:choice>
    <xs:element name="CallReturn" type="EmptyType" />
    <xs:element name="NewCall" type="EmptyType" />
    <xs:element name="FwdAll" type="EmptyType" />
    <xs:element name="CancelFwd" type="EmptyType" />
    <xs:element name="SignIn" type="EmptyType" />
    <xs:element name="SignOut" type="EmptyType" />
    <xs:element name="BackSpace" type="EmptyType" />
    <xs:element name="CANCEL" type="EmptyType" />
    <xs:element name="Redial" type="EmptyType" />
    <xs:element name="MissedCalls" type="EmptyType" />
    <xs:element name="SwitchSCR" type="EmptyType" />
    <xs:element name="ReverseCurrency" type="EmptyType" />
  </xs:choice>
</xs:complexType>
```

```
 <xs:complexType name="SoftKeyType">
  <xs:sequence>
   <xs:element name="Label" minOccurs="0" maxOccurs="1">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
      <xs:maxLength value="32"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:element>
   <xs:element name="Icon" minOccurs="1" maxOccurs="1" type="xs:string">
    <xs:attribute name="x" type="xs:integer"/>
    <xs:attribute name="y" type="xs:integer"/>
    <xs:attribute name="textoffset" type="xs:integer"/>
    <xs:attribute name="isfile" type="xs:boolean"/>
   </xs:element>
   <xs:element name="Action" minOccurs="1" maxOccurs="1" type="IdleActionType">
   </xs:element>
   <xs:element name="displayCondition" type="IdleScreenConditionType" minOccurs="0" maxOccurs="1" nillable="true"/>
  </xs:sequence>
  <xs:attribute name="useshapeid" type="xs:string" use="optional"/>
 </xs:complexType>

</xs:schema>
```

EXAMPLE IDLE SCREEN CUSTOMIZATION

In the XML Idle Screen Customization package, the default xml idle screen files for GXP2100, GXP2110, GXP2120 and GXP1450 are provided. By making slight modification on these default templates, users could customize the logo, softkey and etc based on them. Users will need to change the file name to gs_screen.xml if not use custom file name (refer to section “How it works and Configuration” to use custom filename).

HOW TO CUSTOMIZE LOGO

The following steps show how to customize the idle screen logo by modifying the default xml files in the package. The idle screen logo is Grandstream (gs_logo) if not customized.

1. Find the default xml file for your phone in the package.
2. Due to different screen displays, there are four distinctive sets of “IdleScreen” sections in the xml file. Each section is mapped to the screen by pressing “SwitchSCR” softkey on the phone.
3. Find the first “IdleScreen” section, which is mapped to the first default screen display.
4. Set the <DisplayBitmap isfile=“true”> to <DisplayBitmap isfile=“false”>
5. Users will need to encode the image first. Then insert the encoded image in the appropriate <Bitmap> tag where it is “/app/resource/logo/gs_logo.bmp”. See below:

```
<DisplayBitmap isfile="false">
        <Bitmap>/app/resource/logo/gs_logo.bmp</Bitmap>
                <X>15</X>
                 <Y>26</Y>
        </DisplayBitmap>
```

6. Load the custom idle screen either via TFTP/HTTP in the Web GUI->Advance Settings, then update and reboot.
7. Once image is loaded, the account settings will still appear on the left hand side. This is also customizable if user doesn’t want the left pane appearing.
8. User can also reconfigure the X, Y values from the template to adjust the logo.

HOW TO CUSTOMIZE SOFTKEY

There are four idle screen sections in the xml file with each screen has different softkeys. Users could edit the softkeys in each <IdleScreen> section by modifying the contents within <Softkeys></Softkeys>. The following steps show how to hide the "Redial" softkey in the screen.

1. Find the default xml file for your phone in the package.
2. Due to different screen displays, there are four distinctive sets of "IdleScreen" sections in the xml file. Each section is mapped to the screen by pressing "SwitchSCR" softkey on the phone.
3. Find the first "IdleScreen" section, which is mapped to the first default screen display.
4. In this "IdleScreen" section, find the "Softkeys" section. Remove the contents for Redial softkey. See below:

```
<SoftKey>
 <Action>
   <Redial/>
 </Action>
   <displayCondition>
      <conditionType>hasDialedCalllog</conditionType>
   </displayCondition>
</SoftKey>
```

5. Load the custom idle screen either via TFTP/HTTP.
6. After the Redial is disabled, end users will not be able to redial the number on the phone.

HOW TO CUSTOMIZE ACCOUNT STATUS

In each screen display, the account status will be shown in the left by default. Users could hide the account status for the screen as they prefer. The following steps show how to hide the account status in the second screen which is the weather information.

1. Find the default xml file for your phone in the package.
2. Due to different screen displays, there are four distinctive sets of "IdleScreen" sections in the xml file. Each section is mapped to the screen by pressing "SwitchSCR" softkey on the phone.
3. Find the second "IdleScreen" section, which is mapped to the screen display for updated weather information.
4. Fine the contents where it is <ShowStatusLine>true<ShowStatusLine>.
5. Change "true" to "false".
6. Load the custom idle screen either via TFTP/HTTP.
7. After loading the file, press softkey "SwitchSCR" to the second screen display. Users will find the account status is disabled and the display will automatically be adjusted to fit the screen.

Note: For more idle screen customizations, users could edit the xml file according to the properties and rules set for each section in the xsd file provided in this document.

APPENDIX

ROOT ELEMENT "SCREEN"

The XML document has root element called Screen; it contains 2 mandatory sub-elements called LeftStatusBar and IdleScreen.

```
<xs:element name="Screen">
  <xs:complexType>
    <xs:sequence>
       <xs:element name="LeftStatusBar" minOccurs="1" maxOccurs="1" >
```

```
        </xs:element>
        <xs:element name="IdleScreen" minOccurs="1" maxOccurs="4"/>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

ELEMENT "LEFTSTATUSBAR"

This element contains 2 mandatory sub-elements called Layout and Account, which controls the general template for the layout and account status.

```
<xs:element name="LeftStatusBar" minOccurs="1" maxOccurs="1" >
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Layout" type="GeneralLayoutTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
            <xs:element name="Account" type="GeneralTemplateType" minOccurs="1" maxOccurs="1"></xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
```

ELEMENT "IDLESCREEN"

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

```
<xs:element name="IdleScreen" minOccurs="1" maxOccurs="4">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ShowStatusLine" type="xs:boolean" minOccurs="0" maxOccurs="1" default="true"/>
            <xs:element name="DisplayBitmap" type="BitmapType"  minOccurs="0" maxOccurs="unbounded"
nillable="true"/>
            <xs:element name="DisplayString" type="StringType"  minOccurs="0" maxOccurs="unbounded" nillable="true"/>
            <xs:element name="DisplaySet" type="GeneralListHolderType" minOccurs="0" maxOccurs="1" nillable="true"/>
            <xs:element name="SoftKeys" minOccurs="0" maxOccurs="1" nillable="true"/>
          </xs:sequence>
        </xs:complexType>
 </xs:element>
```

Note: By the above grammar, ShowStatusLine. DisplaySet and SoftKeys could be disabled or enabled (appear once). And DisplayBitmap and DisplayString could have any number of instances.

Display Rules: When both DisplayBitmap and DisplayString elements are present, all bitmaps will be rendered before the strings are displayed. When multiple instances of the same type (bitmap/string) are present, they are displayed in the order they appear in the XML and later objects (bitmap/string) may overwrite/corrupt previous objects.

ELEMENT "DISPLAYSTR"

This element contains the string to be displayed.

```
<xs:element name="DisplayStr" type="xs:string" minOccurs="1" maxOccurs="1" />
```

The string can contain dynamic contents. The following list shows the important 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. $I: This variable is replaced with the system IP address.
6. $D: This variable is replaced with the current day of month with leading zero, possible values: 01, 02, …, 31
7. $d: This variable is replaced with the current day of month without leading zero, possible values: 1, 2, …, 31
8. $M: This variable is replaced with the current month in English, possible values: January, February, …, December
9. $o: This variable is replaced with the current month in number with leading zero, possible values: 01, 02, …, 12
10. $n: This variable is replaced with the current month in number without leading zero, possible values: 1, 2, …, 12
11. $Y: This variable is replaced with the current year in 4-digit number, for example: 2006, 2007
12. $y: This variable is replaced with the current year in 2-digit number, for example: 06, 07 …
13. $P: This variable is replaced with the current AM/PM status in upper case, possible values: AM, PM
14. $p: This variable is replaced with the current AM/PM status in lower case, possible values: am, pm
15. $h: This variable is replaced with the current hour of day in 12-hour representation with leading zero, possible values: 01, 02, …, 12
16. $m: This variable is replaced with the current minute of hour with leading zero, possible values: 01, 02, …, 59
17. $s: This variable is replaced with the current second of minute with leading zero, possible values: 01, 02, …, 59

Note: 1. If users need to display the "$" sign, please use "$$" escape sequence.
2. New variables will be added with the firmware update.

GXP21XX/GXP1450 SCREEN/LOGO SIZE

Model	Screen Size (Pixel)	Default gs_logo Size (Pixel)	Default Logo Zone Size (Pixel)
GXP2100	180 x 90	72 x 35	123 x 58
GXP2120	320 x 160	136 x 66	230 x 85
GXP2110	240 x 120	89 x 46	165 x 50
GXP1450	180 x 60	N/A	N/A