XML Based Customizable Screen

Rev 1.1

August 10, 2006

1. Introduction

Starting from release version 1.0.2.X, GXP-2000 supports the 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-2000.

2. 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.
- Idle-Screen XML 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 Path” is set.

This feature does not automatically download the XML file in the path even when activated. The following 2 options are added to the Preference LCD GUI submenu:

Download SCR XML
Erase Custom SCR

User will have to choose to Download SCR XML to start the download process. Once the XML is successfully downloaded it will be parsed and be effective right away. The file should also be saved for future use and should be loaded automatically after reboot.

3. XML Syntax

XSD file

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="Screen">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="IdleScreen" minOccurs="1" maxOccurs="1">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="ShowStatusLine" type="xsd:boolean" minOccurs="1" maxOccurs="1"
                                                                    default="true"/>
       <xsd:element name="DisplayBitmap" minOccurs="0" maxOccurs="unbounded" nillable="true">
        <xsd:complexType>
         <xsd:sequence>
          <!-- We only accept Windows Monochrome Bitmap, max 130x64 pixels encoded by base64
-->
          <xsd:element name="Bitmap" type="xsd:base64Binary" 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:sequence>
         <xsd:attribute name="a1reg" type="xsd:boolean"/>
        </xsd:complexType>
       </xsd:element>
       <xsd:element name="DisplayString" minOccurs="0" maxOccurs="unbounded" nillable="true">
        <xsd:complexType>
         <xsd:sequence>
          <xsd:element name="DisplayStr" 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:sequence>
         <xsd:attribute name="a1reg" type="xsd:boolean"/>
         <xsd:attribute name="font">
          <xsd:simpleType>
           <xsd:restriction base="xsd:string">
            <xsd:enumeration value="f8"/>
            <xsd:enumeration value="f10"/>
            <xsd:enumeration value="f13h"/>
            <xsd:enumeration value="f13b"/>
            <xsd:enumeration value="f16"/>
            <xsd:enumeration value="f16b"/>
            <!-- f18c is a 18 point Comic font  -->
            <xsd:enumeration value="f18c"/>
           </xsd:restriction>
          </xsd:simpleType>
         </xsd:attribute>
         <xsd:attribute name="halign">
          <xsd:simpleType>
           <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Left"/>
            <xsd:enumeration value="Center"/>
            <xsd:enumeration value="Right"/>
           </xsd:restriction>
          </xsd:simpleType>
         </xsd:attribute>
         <xsd:attribute name="valign">
          <xsd:simpleType>
           <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Top"/>
            <xsd:enumeration value="Center"/>
            <xsd:enumeration value="Bottom"/>
           </xsd:restriction>
          </xsd:simpleType>
         </xsd:attribute>
        </xsd:complexType>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
   </xsd:sequence>
```

```
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

4. Example Idle Screen File

```
<?xml version="1.0"?>
<!-- This file creates identical result to GXP-2000 default behavior -->
<Screen>
        <IdleScreen>
          <ShowStatusLine>true</ShowStatusLine>
          <DisplayString font="f8">
                <DisplayStr>$W, $M $d</DisplayStr>
                <X>0</X>
                <Y>0</Y>
          </DisplayString>
          <DisplayString font="f13h" halign="Center" a1reg="false">
                <DisplayStr>$N</DisplayStr>
                <X>65</X>
                <Y>12</Y>
          </DisplayString>
          <DisplayString font="f13b" halign="Center" a1reg="true">
                <DisplayStr>$N</DisplayStr>
                <X>65</X>
                <Y>12</Y>
          </DisplayString>
          <DisplayString font="f13h" halign="Center" a1reg="false">
                <DisplayStr>$X</DisplayStr>
                <X>65</X>
                <Y>26</Y>
          </DisplayString>
          <DisplayString font="f13b" halign="Center" a1reg="true">
                <DisplayStr>$X</DisplayStr>
                <X>65</X>
                <Y>26</Y>
          </DisplayString>
          <DisplayString halign="Center" valign="Bottom">
                <DisplayStr>$I</DisplayStr>
                <X>65</X>
                <Y>48</Y>
          </DisplayString>
        </IdleScreen>
</Screen>
```

5. XML Explanation

Root Element "Screen"

The XML document has root element called `Screen`; it contains exactly 1 sub-element called `IdleScreen`.

```
  <xsd:element name="Screen">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="IdleScreen" type="IdleScreenType"
                     minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

Element "IdleScreenType"

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

```
<xsd:complexType name="IdleScreenType">
  <xsd:sequence>
    <xsd:element name="ShowStatusLine" type="xsd:boolean"
                 minOccurs="1" maxOccurs="1" default="true"/>
    <xsd:element name="DisplayBitmap" type="DisplayBitmapType"
                 minOccurs="0" nillable="true"/>
    <xsd:element name="DisplayString" type="DisplayStringType"
                 minOccurs="0" nillable="true"/>
  </xsd:sequence>
</xsd:complexType>
```

Note: By the above grammar, `ShowStatusLine` must appear exactly once and any number of `DisplayBitmap` and `DisplayString` 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 "ShowStatusLine"

This Boolean element decides if we will display the status bar on the top of the screen. The "Status Line" includes the registration status icon, volume icon, time/date on the right-top corner, and the horizontal separator line.

This element must appear exactly once in IdleScreenType and has a default value of "true.

When set to false, the origin (x-0, y-0) refers to the absolute top-left corner; when set to true, the origin refers to the reference-origin below status line (x-0,y-16). This means when ShowStatusLine is set to true, all y-offsets are shift down for 16 pixels so the status line will not be corrupted or over-written.

Element DisplayBitmap

This element carries the information on how a bitmap is to be rendered on screen. It has three mandatory elements and one optional attribute:

Element Bitmap

This element contains the bitmap encoded by base64

```
<xsd:element name="Bitmap" type="xsd:base64Binary" minOccurs="1" maxOccurs="1"/>
```

Note: We only accept Windows Bitmap (file header begins with 0x424D) that is monochrome (1-bit depth) and not exceeding 130x65 pixels (that's our LCD resolution). Anything not bound to the above restriction is dropped and ignored. You may use Windows Paint to change an existing BMP file to 1-bit depth.

Elements X and Y

This element contains X and Y offsets from the origin that we will use to render the bitmap.

```
<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"/>
```

Attribute a1reg

This OPTIONAL attribute specifies the conditions when the bitmap will be displayed.

```
<xsd:attribute name="a1reg" type="xsd:boolean"/>
```

When this attribute is present and the value is “true” then the bitmap will be displayed ONLY when SIP Account 1 is in REGISTERED state.

When this attribute is present and the value is “false” then the bitmap will be displayed only when SIP Account 1 is NOT in REGISTERED state.

Note: When this attribute is absent then this bitmap is displayed regardless to the SIP Account 1 registration states.

Element DisplayString

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

Element DisplayStr

This element contains the string to be displayed

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

The string can contain dynamic contents. As present, we support the following 18 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 24-hour representation with leading zero, possible values: 00, 02, …, 23
16. $h: This variable is replaced with the current hour of day in 12-hour representation with leading zero, possible values: 01, 02, …, 12
17. $m: This variable is replaced with the current minute of hour with leading zero, possible values: 01, 02, …, 59
18. $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.

Elements X and Y

This element contains X and Y offsets from the origin that we will use to render the string.

```
<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"/>
```

Attribute a1reg

This OPTIONAL attribute specifies the conditions when the string will be displayed.

```
<xsd:attribute name="a1reg" type="xsd:boolean"/>
```

When this attribute is present and the value is “true” then the string will be displayed ONLY when SIP Account 1 is in REGISTERED state.

When this attribute is present and the value is “false” then the string will be displayed only when SIP Account 1 is NOT in REGISTERED state.

Note: When this attribute is absent then this bitmap is displayed regardless to the SIP Account 1 registration states.

Attribute font

This OPTIONAL attribute specifies the font we will use to render the string.

```
<xsd:attribute name="font" type="fontType"/>

<xsd:simpleType name="fontType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="f8"/>
```

```
      <xsd:enumeration value="f10"/>
      <xsd:enumeration value="f13h"/>
      <xsd:enumeration value="f13b"/>
      <xsd:enumeration value="f16"/>
      <xsd:enumeration value="f16b"/>
      <!-- f18c is a 18 point Comic font  -->
      <xsd:enumeration value="f18c"/>
    </xsd:restriction>
</xsd:simpleType>
```

In this version we will support 1 system font and 7 additional fonts in various sizes as enumerated above. Any fonts not recognize will default to the system font. When this attribute is absent then we will also default to system font.

Attribute halign

This OPTIONAL attribute specifies the horizontal alignment method used to display the string.

```
<xsd:attribute name="halign" type="HorizontalAlignmentType"/>

<xsd:simpleType name="HorizontalAlignmentType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Left"/>
      <xsd:enumeration value="Center"/>
      <xsd:enumeration value="Right"/>
    </xsd:restriction>
</xsd:simpleType>
```

We will default to Left when this attribute is absent.

Note: When using the Center alignment you will need to calculate the midpoint for the x-coordinate (4.5.2) for the width to be considered centered. For instance, element X must be set to 65 (130/2) to display a string that is aligned to the center of the LCD. Similarly, you will need to specify the right most point to render if you are using the Right halign method.

Attribute valign

This OPTIONAL attribute specifies the vertical alignment method used to display the string.

```
<xsd:attribute name="valign" type="VerticalAlignmentType"/>

<xsd:simpleType name="VerticalAlignmentType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Top"/>
      <xsd:enumeration value="Center"/>
      <xsd:enumeration value="Bottom"/>
    </xsd:restriction>
</xsd:simpleType>
```

We will default to Top when this attribute is absent.

Note: When using the Center alignment you will need to calculate the midpoint for the y-coordinate (4.5.3) for the width to be considered centered. For instance, element Y must be set to 32 to display a string that is aligned to the center of the LCD. Similarly,

you will need to specify the right most point to render if you are using the Bottom valign method (set to 64).