

Grandstream Networks, Inc.

XML Based Idle Screen Customization Guide

GRP260X IP Phone



GRP2601(P)



GRP2602(P/W/G)



GRP2603(P)



GRP2604(P)

INTRODUCTION

The Grandstream GRP260X supports XML based idle screen customization. This XML API allows users to customize the idle screen layout as well as the logo, text or system variables to be displayed. The design of the display and layout depends highly upon personal preferences and requirements.

This document specifies the Grandstream XML Customizable Screen API design that will be used on GRP260X.

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, and etc.). Almost all documents have certain kind of structure.

*Note: 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 where the phones could interact with external applications in a flexible and programmable manner. Two specific XML APIs supported by GRP260X are XML Custom Screen and XML Phonebook.

XML API ARCHITECTURE

The XML idle screen customization API on GRP260X could use HTTP/HTTPS or TFTP as the transport protocol. The following figure shows how it works via HTTP as an example. Basically, GRP260X initiates the HTTP GET Request to the HTTP server and waits for the response. Once the phone receives the response with XML content in BODY, it displays the information.

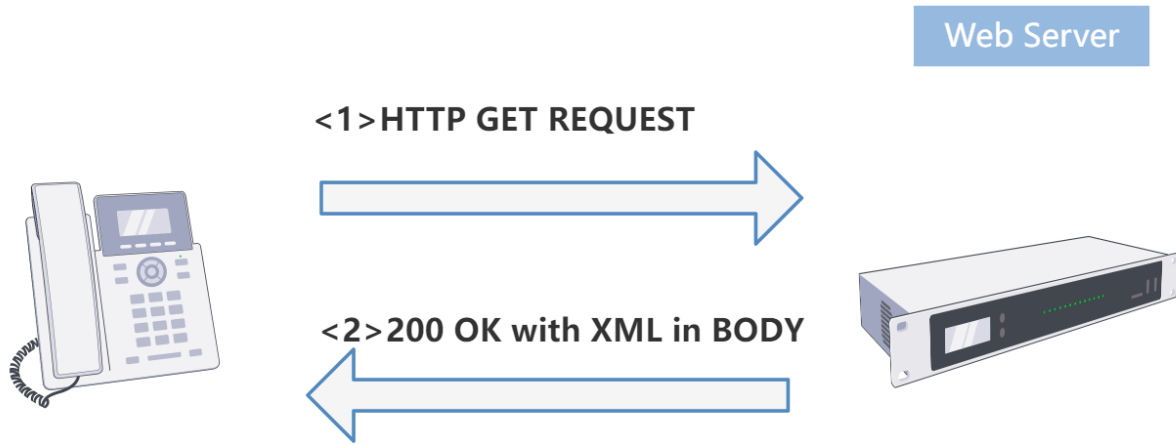


Figure 1: GRP260X XML API via HTTP

Two types of XML API architectures are introduced below, depending on whether the transaction is within a LAN or accessed via the Internet.

1. A transaction in LAN area may exchange information in the following manner. GRP260X sends request and accepts XML contents via HTTP/HTTPS/TFTP, directly communicating with the HTTP/HTTPS/TFTP 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 idle screen display. The following figure shows downloading XML idle screen via HTTP within LAN.

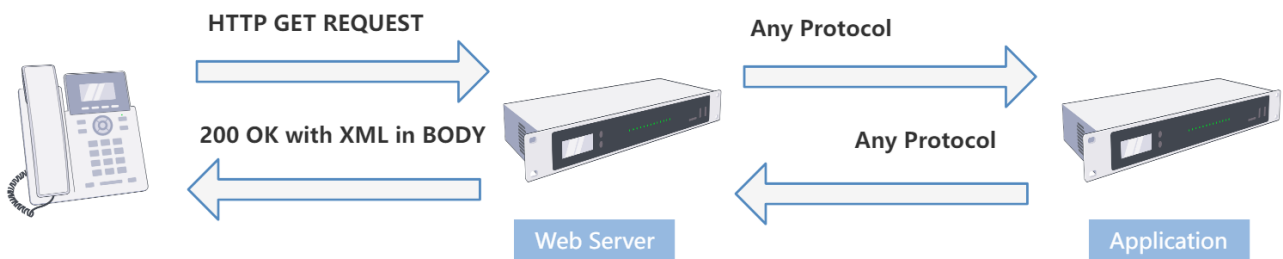


Figure 2: GRP260X XML API Structure - 1

2. If the above Web Server accesses Internet, it could interact with outside web server and respond real-time content to GRP260X.

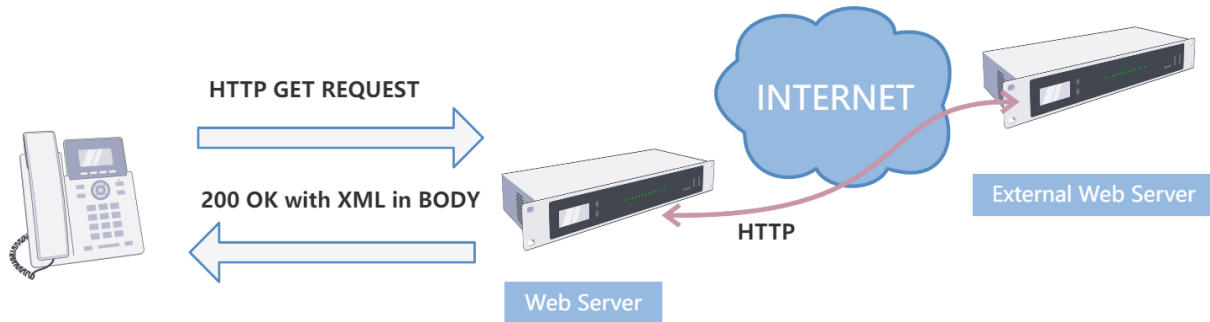


Figure 3: GRP260X XML API Structure - 2

As illustrated above, all the logic lies within the server side of the architecture. The GRP260X phone receives XML document and displays it accordingly.

XML IDLE SCREEN DOWNLOADING CONFIGURATION

FIRMWARE

Before the XML idle screen customization is used on GRP260X, please make sure the firmware on the phone is upgraded to the latest version. Please refer to the following link for firmware upgrading information:

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

GRP260X Graphic Display

The following table lists all the GRP260x screen sizes.

Table 1: GRP260X Idle Graphic Display

Model	Graphic Display
GRP2601/GRP2601P	132 x 48 (2.21' ') LCD display
GRP2602/GRP2602P/GRP2602W	132 x 48 (2.21' ') backlit graphical LCD display
GRP2603/GRP2603P	132 x 64 backlit graphical LCD display
GRP2604/GRP2604P	132 x 64 backlit graphical LCD display

WEB GUI CONFIGURATION

To download the XML idle screen file to GRP260X, firstly enable the Idle Screen XML Download and configure the Server Path under Web GUI->Settings->XML Applications page.

XML Application

Idle Screen

Idle Screen XML Download ⓘ	Enabled, use HTTP ▾
Download Screen XML at Boot-up ⓘ	<input checked="" type="checkbox"/>
Use Custom Filename ⓘ	<input type="checkbox"/>
Idle Screen XML Server Path ⓘ	192.168.160.136:80/folder
XML File Example ⓘ	Download

[Save](#) [Save and Apply](#) [Reset](#)

Figure 4: Web GUI Configuration

- **Idle Screen XML Download**

The idle screen XML file could be downloaded via HTTP, HTTPS or TFTP. By default, it is "Disabled".

- **Download Screen XML at Boot-up**

If it's set to "Yes", when the phone boots up, it will send out request to download the XML idle screen file automatically.

If it's set to "No", users would need go to LCD MENU->Preference->Download SCR XML to download the idle screen manually. Users could also erase the current downloaded idle screen by pressing MENU->Preference->Erase Custom SCR. By default, this option is set to "No".

- **Use Custom Filename**

If "Use custom filename" is set to "No", the XML file name must be **idle_screen.xml**. In this case, users only need to specify the folder path in "Idle Screen XML Server Path" where the idle_screen.xml is located (For example, 192.168.40.10/XMLscreendir).

If "Use custom filename" is set to "Yes", users could name the file as preferred. In this case, the "Idle Screen XML Server Path" has to be specified to the name of the XML file (For example, 192.168.40.10/XMLscreendir/welcome.xml).

- **Idle Screen XML Server Path**

This specifies the path where the phone is going to download the XML file. Since downloading method is selected as "HTTP", "TFTP", or "HTTPS" already when enabling the XML idle screen download, users shall not specify the protocol in this path anymore. The accepted format are as follows:

IP_address[:port]/dir

IP_address[:port]/dir/filename

Hostname[:port]/dir

Hostname[:port]/dir/filename

Examples:

192.168.40.10/XMLscreendir

192.168.40.10/XMLscreendir/welcome.xml

192.168.40.10:443/XMLscreendir

192.168.40.10:443/XMLscreendir/welcome.xml

mycompany.com/gs_screen_dir

mycompany.com:8080/idlescreen

service.mycompany.com/XML/GRP260Xidle.xml

service.mycompany.com:8080/XML/GRP260Xidle.xml

Note:

- If "[:port]" is not specified, port 80 will be used as default for HTTP; port 443 will be used as default for HTTPS; port 69 will be used as default for TFTP;
 - If "Use custom filename" is set to "No", the Server Path does not necessarily need to contain the destination file name. Users only need specify the directory path where the file is located.
-

After the above configuration, click on "Save and Apply" in the web GUI page. The phone will apply and display the XML idle screen upon pressing MENU->Preference->Download XML SCR in phone's LCD or once the phone restarted if the option "Download Screen XML at Boot-up" is selected.

Users may also use the XML configuration file to provision the phone with the above XML idle screen downloading options. In this case, GRP260X needs to be rebooted and provisioned. The corresponding P values are as below.


- **P340:** Idle Screen XML Download. Possible values: 0 (Disabled) / 1 (HTTP) / 2 (TFTP) / 3 (HTTPS), other values are ignored;
- **P1349:** Download Screen XML at Boot-up. Possible values: 0 (No) / 1 (Yes), other values are ignored;
- **P1343:** Use custom filename. Possible values: 0 (No) / 1 (Yes), other values are ignored;
- **P341:** Idle Screen XML Server Path. This is a string up to 128 characters.

Note:

- As you may know, it is also possible to enter the idle screen file server path into a web browser. In this way you'll be able to see the exact XML document on your PC that your phone will be receiving;
 - The downloaded XML idle screen file can be displayed but won't be saved internally on the phone. It is recommended to save the XML idle screen file externally on your PC or server for your reference and future use.
-

GRP260X IDLE SCREEN

OVERVIEW

Without XML idle screen customization or other configurations, a GRP260X's default idle screen is like below. Press the  softkey will toggle among the following different idle screens (vary among models):

- Default idle screen when the phone boots up
- IP Address and Extension Number



Figure 5: GRP260X Default Idle Screen

The idle screens for all models are specified in the table below. Although all the supported idle screens could be modified via the XML idle screen file, normally the idle screen 1 (default idle screen) is the only one user would need to customize for logo display, text display and etc.

Table 2: GRP260X Idle Screens Overview

Model	Idle Screen 1	Idle Screen 2
GRP2601/GRP2601P	Default	IP Address
GRP2602/GRP2602P/GRP2602W	Default	IP Address
GRP2603/GRP2603P	Default	IP Address
GRP2604/GRP2604P	Default	IP Address

XML DOCUMENT 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 GRP260X, 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.

Table 3: Special Characters in XML Document

Characters	Name	Escape Sequence
&	Ampersand	&
"	Quote	"
'	Apostrophe	&apos
<	Left Angle Bracket	<
>	Right Angle Bracket	>

GRP260X SCREEN XML STRUCTURE

The main structure of GRP260X idle screen template could be represented below. This provides users with an overview of the XML element and screen layout. For more details on element attributes and text information, please refer to section [GRP260X IDLE SCREEN XML ELEMENT].

GRP260X SCREEN XML MAIN STRUCTURE

```
<Screens>
  <Screen model="GRP2603;GRP2604" width="132" height="64">
    <!-- MAIN SCREEN->
    <IdleScreen>
    </IdleScreen>

    <IdleScreen>
    </IdleScreen>

    <IdleScreen>
      <!-- DEFINE AVAILABLE SOFTEKEYS HERE->
    </IdleScreen>
    <!-- DISPLAY IP ADDRESS AND EXTENSIONS->
    <IdleScreen>
    </IdleScreen>
  </Screen>

  <Screen model="GRP2601,GRP2602" width="132" height="48">
    <!-- MAIN SCREEN->
    <IdleScreen>
    </IdleScreen>

    <IdleScreen>
    </IdleScreen>

    <IdleScreen>
      <!-- DEFINE AVAILABLE SOFTEKEYS HERE->
    </IdleScreen>
    <!-- DISPLAY IP ADDRESS AND EXTENSIONS->
    <IdleScreen>
    </IdleScreen>
  </Screen>
</Screens>
```

GRP260X IDLE SCREEN XML ELEMENT

This section describes details of the XML element used in GRP260X XML idle screen customization. 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. This element is mandatory.

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

<Screen> ELEMENT DETAILS

The following tables shows child element and attribute for <Screen> element.

Table 4: <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
IdleScreen	Child element	Mandatory	-	Main customization area. Normally, only the 1st <IdleScreen> element needs customization for logo, text or variable display. Users usually could leave the other IdleScreen elements as default.

<IdleScreen> ELEMENT

This is the main customization section for the XML idle screen.

```

<IdleScreen>
  <ShowStatusLine> true/false </ShowStatusLine>
  <DisplayElement>
    Element display information here
  </DisplayElement>
  <DisplayBitmap >
  <DisplayString font="unifont">
  </DisplayString>
  <SoftKeys>
    Softkey
  </SoftKeys>
</IdleScreen>

```

<IdleScreen> ELEMENT DETAILS

Table 5: <IdleScreen> Element

Object	Position	Type	Values	Comments
IdleScreen	Element	Mandatory	-	
ShowStatusLine	Child element	Mandatory	-	It could use "true" or "false" as its text. "true": the line label on the left side will always display. "false": the line label on the left side will not display. For the 1st <IdleScreen> section (default screen), it should be set to "true".
DisplayElement	Child element	Optional	-	
DisplayBitmap	Child element	Optional	-	
DisplayString	Child element	Optional	-	
Softkeys	Child element	Mandatory	-	

<DisplayElement> ELEMENT

This element contains all the contents to be displayed, i.e., string, picture and rectangle.

```

<DisplayElement>
  <DisplayString>String information</DisplayString>
  <DisplayBitmap>Image information</DisplayBitmap>
  <DisplayRectangle x="X location" y="Y location" width="Width" height="Height"
  bgcolor="Background color"/>
</DisplayElement>

```

<DisplayElement> ELEMENT DETAILS

Table 6: <DisplayElement> Element

Object	Position	Type	Values	Comments
DisplayElement	Element	Mandatory	-	
DisplayString	Child element	Optional	-	Displays string
DisplayBitmap	Child element	Optional	-	Displays bitmap picture
DisplayRectangle	Child element	Optional	-	Displays rectangle

<DisplayString> ELEMENT

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

```

<DisplayString font =" unifont/bold" 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> ELEMENT DETAILS

Table 7: <DisplayString> Element

Object	Position	Type	Values	Comments
DisplayString	Element	Optional	-	
font	<DisplayString> Attribute	Optional	"unifont" /"bold"/"time" /"time_14"/"time14_bold"	Default font type is "unifont". time/time_14/time_14_bold:It used to display the time
width	<DisplayString> Attribute	Optional	int	
height	<DisplayString> Attribute	Optional	int	
align	<DisplayString> Attribute	Optional	string	Default value is "left"
color	<DisplayString> Attribute	Optional	string	Default value is "Black" Only if set to White is White, all other colors are black.
bgcolor	<DisplayString> Attribute	Optional	string	Default value is "White" Only if set to White is White, all other colors are black.
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. System variables can be used here
displayCondition	Child element	Optional	-	The string will be displayed under certain condition. If not specified it's always displayed

<DisplayBitmap> ELEMENT

This element is to display a bitmap picture in the screen, for example, to customize the logo. Inside the <Bitmap> tag, the picture must be encoded in base 64 format already. If you search "Base 64 Encoder" online, there are plenty of online tools as base 64 encoder to encode the .bmp picture.

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

<DisplayBitmap> ELEMENT DETAILS

Table 8: <DisplayBitmap> Element

Object	Position	Type	Values	Comments
DisplayBitmap	Element	Optional	-	
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.
- Use a base-64 encoder to encode the picture.
- Copy and paste and encoded result inside <Bitmap> tag.
- Make sure the isfile attribute is set to "false" in <DisplayBitmap> element.

Example:

```

<DisplayBitmap >
  <Bitmap>Qk3GAgAAAAAAD4AAAAoAAAAqgAAABsAAAABAAEAAAAAAIgcCAAAAAAAA
  AAAAAAAAAAAAAAAAAAAP///wD////////////////////////////////AAAD////////////////////////////////AAAD////////////////////////////////
  AAAD+D/wf/wAAwAfgAB8B/wPw/wA+AAHAAAD+B/wH8AAAAAAAAAAAH4DwPgAAAABAAAD+A/w
  D4H//wPwH//g4D4BwDgfgf//AAAD+A/wB4H//wHwP//B8D4AwBgPgf//AAAD+Afwb4H//wHwP//B8B8AY
  AwPgf//AAAD/APwB4D4HwHwHx/A8B8CAEAPgPAfAAAD/AHwA8D4H4D4H4gfg8B+BgCAHgPAfAAA
  D/ADwA8B8H4D4DgfgD+BwDgHgHgAAAD/ABwEfgAP4B/AAfwAAAA8D4H8AA/AAAD/AgwEPwA/8
  B/wB/+AAAA/D8H/AD/AAAD/AwQGP///8B////////////////////////////////AAAD/A4QGH///8B////////////////////////////////AAAD/AcAGD///+B
  //////////////////////////////////AAAD/geAHB///+B////////////////////////////////AAAD/geAHh///+A////////////////////////////////AAAD/gfgHw////A////////////////////////////////AAA
  D/gfgHwf//A////////////////////////////////AAAAAAfwH4f//gf////////////////////////////////AAACAAfwH4P//gf////////////////////////////////AAADAA/4H8P////////
  //////////////////////////////////AAADgB/8H+P////////////////////////////////AAAD////////////////////////////////AAAD////////////////////////////////AAAD////////////////////////////////
  //////////AAAA= </Bitmap>
    <X>0</X>
    <Y>7</Y>
</DisplayBitmap>
  
```

<DisplayRectangle> ELEMENT

This element is to render rectangle display. It could be used as a frame or background bar.

```

<DisplayRectangle x="X location" y="Y location" width="Width" height="Height"
bgcolor="Background color" border-color="Rectangle border color"/>
  
```

<DisplayRectangle> ELEMENT DETAILS

Table 9: <DisplayRectangle> Element

Object	Position	Type	Values	Comments
DisplayRectangle	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. Only if set to White is White, all other colors are black.
border-color	<DisplayRectangle> Attribute	Optional	string	Default value is None. Only if set to White is White, all other colors are black.

<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> ELEMENT DETAILS

Table 10: <Softkeys> Element

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. The text for <Action> and <conditionType> are pre-defined in the firmware already so it's recommended to keep this section by default.

```

<SoftKey>
  <Action>
    Pre-defined softkey actions here
  </Action>
  <displayCondition>
    <conditionType>Pre-defined condition Type here</conditionType>
  </displayCondition>
</SoftKey>

```

A new action type "Dial" is added with a different format. To configure a speed dial softkey, user can add a "Dial" softkey as a child element of <SoftKeys> element.

```
<SoftKey action="Dial" label="label name" commandId="Account index" commandArgs="The number to dial"/>
```

Note: the object <label> defines the softkey display name on LCD; <commandId> specifies the account index to dial out the call from, starting from 0 for account 1; <commandArgs> specifies the phone number to dial.

<SoftKey> ELEMENT DETAILS

Table 11: <SoftKey> Element

Object	Position	Type	Values	Comments
SoftKey	Element	Mandatory	-	
Action	Child Element	Mandatory	-	Softkey Action (pre-defined)
displayCondition	Child Element	Mandatory	-	Softkey display condition (pre-defined)

Lists of the pre-defined softkey <Action> and <ConditionType> contents are described below. Please refer to full ConditionType listed in section "XML Idle Screen Display Condition Type".


Table 12: Pre-defined Softkey <Action> and <ConditionType>

Softkey	Action	ConditionType	Description
SwitchSCR	<SwitchSCR/>	SubScreen	To switch among default idle screen and IP address screen
Calls	<Calls/>		To display the records of call.
MissedCalls	<MissedCalls/>	missCall	Displayed when there is new missed call.
OutgoingCalls	<OutgoingCalls>		To display the records of outgoing call.
IncomingCalls	<IncomingCalls>		To display the records of incoming call.
FwdedCalls	<FwdedCalls/>	hasFowardedCallLog	Displayed when account1 is registered and "Enable Call Feature" is set to "Yes"
FwdAll	<FwdAll/>	callFwdCancelled	Displayed when Account 1 is registered, "Enable Call Feature" is set to "Yes," ForwardALL softkey was not set as hidden and Account 1 forwardAll is not set
CancelFwd	<CancelFwd/>	callFwded	Displayed when account1 has Call Forward All activated
Redial	<Redial/>	hasDialedCallog	Displayed when there is dialed call

VoiceMail	<VoiceMail/>	By default it's not specified and it will be always displayed. It could use "hasVoiceMail" so it will be displayed only when there is new voicemail.	To display Voicemail softkey.
Phonebook	<PhoneBook/>	If not specified, it will be always displayed	To bring up phonebook entries
BSCallCenter	<BSCallCenter/>	bsCallCenter	Displayed when Broadsoft Call Center is configured
Call park	<CallParked/>	hasBWCallParks	Display when Broadsoft Call park is configured
LDAP	<LDAP/>	LDAPConfigured	Displayed when LDAP is configured
Dial	<Dial>		Speed dial
Login	</Login>	login	Public mode Login
Logout	</Logout>	logout	Public mode Logout

Example 1: Speed dial

```
<SoftKeys>
  <SoftKey action="Dial" label="SpeedDial" commandId="0" commandArgs="1002"/>
</Softkeys>
```

Note: The order of softkey displayed on the idle screen follows the same order of the <SoftKey> tag defined in the idle screen xml file. For example, in idle_screen.xml file, the action of the first four SoftKey are <SwitchSCR/> , "Dial" , "Calls" and "PhoneBook" . By loading this xml file, the softkeys of  , Dial and Calls and PhoneBook will be placed on phone's first idle screen, in turn.

SYSTEM VARIABLES IN STRING DISPLAY

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

Table 13: System Variables for XML Idle Screen

\$String			
\$a	This variable is replaced with the configured account name	\$B	This variable is replaced with the current day of month with leading zero, possible values: 01, 02, ..., 31
\$b	N/A	\$C	This variable is replaced with DND (Do-Not-Disturb) label when DND is enabled
\$c	This variable is replaced with Missed Call string along with missed call count.	\$D	This variable is replaced with the current day of month with leading zero, possible values: 01, 02, ..., 31
\$d	This variable is replaced with the current day of month with leading zero, possible values: 1, 2, ..., 31	\$E	N/A
\$f	This variable is replaced with the Month-week-date format based on the configuration	\$F	N/A
\$g	N/A	\$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

\$k	This variable is replaced with "Keypad is locked"	\$J	N/A
\$l	N/A	\$K	N/A
\$m	This variable is replaced with the current minute of hour with leading zero, possible values: 01, 02, ..., 59	\$L	N/A
\$n	This variable is replaced with the current month in number with leading zero, possible values: 1, 2, ..., 12	\$M	This variable is replaced with the current month in English, possible values: January, February, ..., December
\$o	This variable is replaced with the current month in number with leading zero, possible values: 01, 02, ..., 12	\$N	This variable is replaced with the configured SIP Display Name or account name
\$p	N/A	\$O	N/A
\$r	This variable is replaced with the volume level	\$P	This variable is replaced with the current AM/PM status in upper case, possible values: AM, PM
\$s	This variable is replaced with the current second of minute with leading zero, possible values: 01, 02, ..., 59	\$R	N/A
\$t	N/A	\$S	N/A
\$v	This variable is replaced with 5V power usage alert message when incorrect power is used	\$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
\$w	N/A	\$V	This variable is replaced with the configured Account SIP Server host

\$x	N/A	\$W	This variable is replaced with the current day of week and has the following possible values: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
\$y	This variable is replaced with the current year in 2-digit number, for example: 06, 07	\$X	This variable is replaced with the configured Account SIP User ID
\$Y	This variable is replaced with the current year in 4-digit number, for example: 2006, 2007 ...	\$-O	This variable is replaced with the account name of the first registered account

Table 14: System Variables for XML Idle Screen (\$+number)

\$+number	This variable is replaced with Multi-language.Number indicates the translation ID to be obtained	\$+161	This variable is replaced with the IP address
\$+640	This variable is replaced with the "NETWORK STARTING" message	\$+339	This variable is replaced with the Account Name
\$+1512	This variable is replaced with the "Enable Auto Recovery" message	\$+1226	This variable is replaced with the "NETWORK DOWN" message
\$+1539	This variable is replaced with the "New IM(s)" message		

Note:

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

To display "\$T", Some fonts contain "AM/PM", so you don't need an extra "\$P". if the font is set to "time" or "time_12", it will display with "AM/PM" , so you need to remove "\$P".And if is set to "time_14" , you need to add "\$P".

XML IDLE SCREEN DISPLAY CONDITION TYPE

The following tables list all the available <ConditionType> in XML idle screen. They could be used for softkey display or string display.

Table 15: ConditionType for XML Idle Screen

ConditionType	Description for softkey/string display
SubScreen	Displayed on idle screen and IP address screen.
missCall	Displayed when there is new missed call
hasFowardedCallLog	Displayed when account1 is registered and "Enable Call Feature" is set to "Yes"
callFwdCancelled	Displayed when Account 1 is registered, "Enable Call Feature" is set to "Yes," ForwardALL softkey was not set as hidden and Account 1 forwardAll is not set
callFwded	Displayed when account1 has Call Forward All activated
hasDialedCallog	Displayed when there is dialed call
hasVoiceMail	Displayed when there is new voicemail.
alwaysDisplay	Default display if not specified
bsCallCenter	Valid if Broadsoft call center is enabled
hasBWCallParks	Valid if Broadsoft Callpark is configured in any of the accounts
LDAPConfigured	Valid if LDAP server is configured
keypadLock	Valid if keypad is currently locked
networkUp	Valid when the phone obtains an IP address (Either IPv4 or IPv6)
networkStart	Valid when the phone obtains an IPv4 address
wrongPower	Valid when the phone used an incorrect power adapter
hasIM	Valid if there is a new instant message in Account 1
login	Public mode Login
logout	Public mode Logout

XML IDLE SCREEN EXAMPLE

GRP260X COMPANY NAME EXAMPLE (/custom_name_GRP260X)

In this example, the company name is added in the middle of the screen. The date element is displayed at the top right and the time at the top left. The second screen shows IP and account1 information.



Figure 6: XML Idle Screen Example - Company name (GRP2602W)



Figure 7: XML Idle Screen Example - Company name (GRP2604P)

GRP260X COMPANY LOGO EXAMPLE (/custom_logo_GRP260X)

In this example, the company logo is added in the middle of the screen. The date element is displayed at the top right and the time at the top left. The second screen shows IP and account1 information.

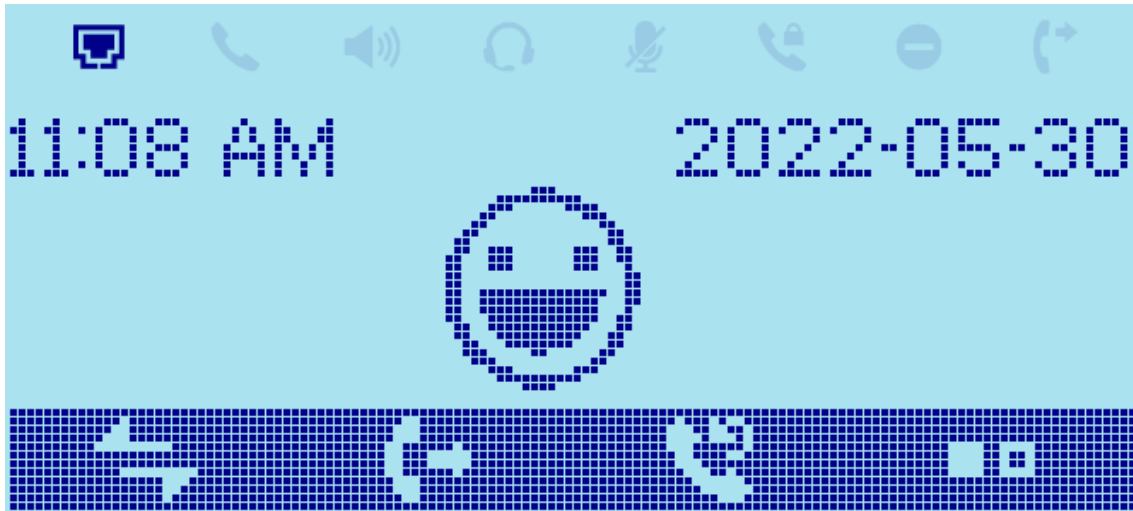


Figure 8: XML Idle Screen Example - Company Logo (GRP2602W)

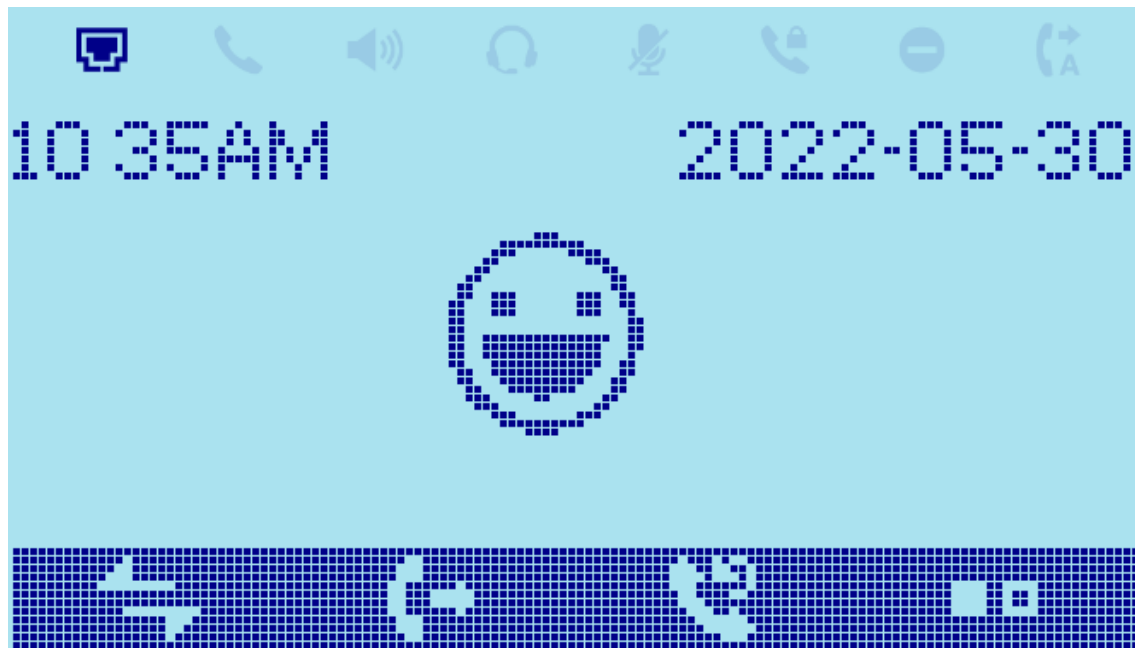


Figure 9: XML Idle Screen Example - Company Logo (GRP2604P)

GRP260X CUSTOM SOFTKEY EXAMPLE (/custom_softkey_GRP2602X)

In this example, a custom softkey labeled "OutgoingCall" and "PhoneBook" are added to the softkey bar. The network status, date and time elements remain as default. The second screen shows IP and account1 information.

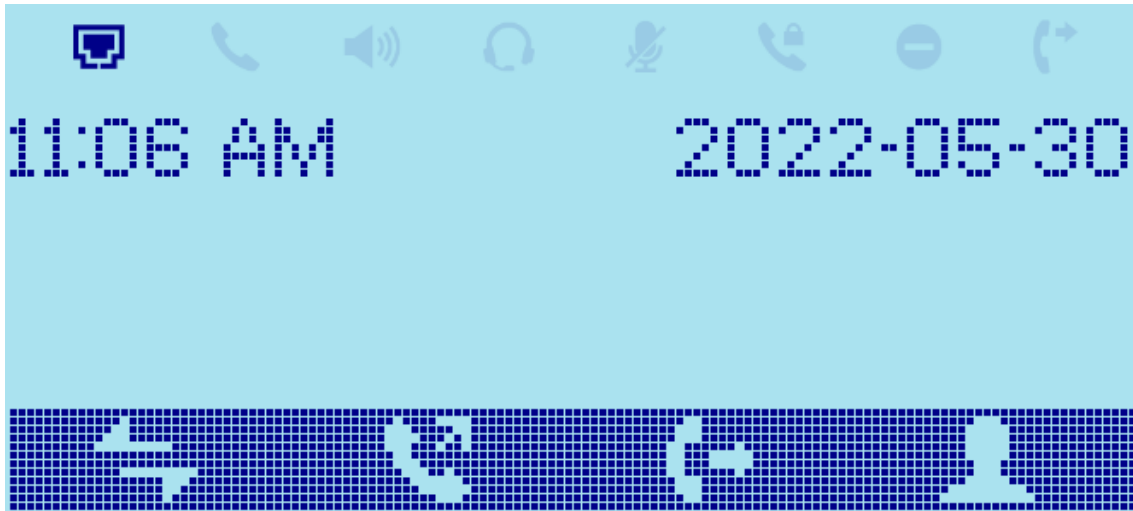


Figure 10: XML Idle Screen Example - Custom softkey (GRP2602W)